Some insights about the Uncapacitated Examination Timetabling Problem

Christos Gogos*, Angelos Dimitsas*, Nastos Vasileios*, and Christos Valouxis†
*Dept. of Informatics and Telecommunications University of Ioannina, Arta, Greece 47100
Email: {cgogos, pint00082, thl1715876}@uoi.gr
†Dept. of Electrical and Computer Engineering, University of Patras, Greece 26500
Email: cvalouxis@upatras.gr

***Abstract*—Timetabling problems easily become hard due to their inherent combinatorial structure. In this work we investigate the examination timetabling problem for Universities. We step back and analyze the best known variant of the problem, the Uncapacitated Examination Timetabling Problem (UETP). In the process we unmask some properties of the UETP. Firstly, we detect components that can be used for decomposition of problem instances by running a connected components identification algorithm. This process reveals the presence of more of the so called noise students and noise examinations that can safely be removed from the problem without affecting the essence of the problem. Secondly, we revisit the idea of identifying lower bounds based on penalties that are impossible to avoid. We strengthen those bounds by exploiting the theoretically maximum number of students that could possibly be scheduled in a single period. Thirdly, we propose a novel way of breaking symmetry, based on ordering examinations that share common characteristics. Since examination timetabling is an active research problem, counting several contributions each year, the present work aims to serve as a useful resource, by revealing properties of the UETP that were not identified before.**

***Index Terms*—scheduling, bounds, examination timetabling, symmetry breaking**

I. Introduction

The Uncapacitated Examination Timetabling Problem (UETP) is a classic timetabling problem. The task is to schedule examinations in available periods such as that no student has to take more than one examination in each period. Moreover, the resulting timetable should have enough distances among examinations for all students so as to promote better preparation and less anxiety. Carter et al. [1], provided 13 real world instances which became known as the Carter datasets and are since used frequently as benchmarks. In this work we examine the UETP and contribute some insights about the Carter datasets.

The paper is organized as follows. The next section succinctly presents the rather extended related work about examination timetabling. Section III describes the specific variation of the problem considered in this work and analyzes the benchmark datasets. Section IV describes an improved way of computing lower bounds. Section V identifies symmetries that exist in UETP. In particular certain examinations can be characterized as interchangeable with other examinations in the final schedule, thus enabling symmetry breaking. Section VI describes mathematical formulations that can result to implementations used as "big" moves in a greater search schema of locating advantageous solutions. Finally, the last section presents conclusions and our plans for further work.

978-1-6654-2742-5/21/$31.00 ©2021 IEEE

II. Related Work

The examination timetabling problem is a well known and much studied combinatorial optimization problem. A survey paper that can be consulted to "get a feeling" about the problem is the work of Qu et al. [2]. A variant of the problem that has drawn much attention is the Uncapacitated Examination Timetabling Problem (UETP), which was proposed by Carter back in 1996 [1], and is considered in this work. In this context uncapacitated means that no room availabilities are considered. Another, more recent survey paper that focus on the UETP is the work of Aldeeb et al. [3].

Several solution approaches have been proposed through the last 25 years. They can be broadly categorized to heuristics, metaheuristics, exact solvers and hybrids. A short description of these approaches is given bellow, alongside with a far from exhaustive set of relevant references. Heuristic approaches are usually based on the resemblance of the problem to graph coloring [4]. Metaheuristics is a broad category including simulated annealing, evolutionary algorithms, nature inspired algorithms and others [5], [6]. This category represents the majority of approaches for the UETP, especially during recent years. Exact solvers mainly includes solvers for Linear and Mixed Integer Programming, but also Constraint Programming solvers, SAT solvers and others [7]. In practice, it has been observed that UETP is "out of reach" for current, state of the art, exact solvers for problem sizes of practical interest [8]. So, usually some form of problem decomposition is employed when exact solvers are used [9]. Finally, many hybrid approaches have been attempted that managed to combine, at some level, advantages of their constituent techniques [10].

Our study of related work concludes to the following points. Examination timetabling and UETP in particular is an active field of research. UETP is a "clean" (simple, easy to state) problem, alongside with datasets that can be used as a testbed for various approaches. Nevertheless, some research use variants of the datasets with the same names and produce difficult to compare results [2]. Thankfully the recent work

of Bellio et al. [8] seems to settle the situation. Most of the research focus on providing better than the previously best solutions. Nevertheless, implementation details and computing effort may be the cause of one approach outperforming the other. No theoretical proven results exist for a set of problems that are over 25 years old [1].

III. Problem Description

UETP problem instances are sets of examinations with each set representing all the examinations taken by a student. Each instance can be perceived as an undirected weighted graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ where vertices $\mathbb{V}$ represent examinations and edges $\mathbb{E}$ represent common students between examinations. The weight of an edge is the number of common students between the two examinations that reside at the ends of the edge. The only imposed hard constraints are that a) all examinations should be scheduled once and that b) no student should be allowed to participate in more than one examination per period. As already mentioned, uncapacitated means that no constraints exist regarding rooms. Likewise, no other constraints that are common in real life examination timetable problems exist. Such constraints are examiners' availabilities, order among examinations, grouping of periods in week days and others. Therefore, UETP can be considered as an abstraction of the real life examination timetabling problem.

The quality of a timetable is measured by an objective function that penalizes the uneven distribution of examinations added for all students. In particular, each student imposes a penalty of 16, 8, 4, 2, 1 for distances of 1, 2, 3, 4, 5 periods between every two of his examinations. Finally, the objective value is normalized by the total number of students.

A. UETP formulation terms

In this section, we define terms that are used throughout later parts of the paper. Set $\mathbb{S}$ is the set of students, set $\mathbb{X}$ is the set of examinations and set $\mathbb{P}$ is the set of periods, arranged in $1..P$ consecutive time-slots. For each student $s \in \mathbb{S}$, $\mathbb{X}_s$ is the subset of examinations that student s is enrolled to. As mentioned before $\mathbb{V}$ is the set of vertices and $\mathbb{E}$ is the set of edges of the corresponding graph $\mathbb{G}$. The number of common students between two examinations $x_i, x_j \in \mathbb{X}$ is given by w_{x_i,x_j}, while the number of enrolled students in each examination x_i is given by r_{x_i}. Finally, b_{x_i} is the set of examinations that have common students with examination x_i.

B. Analysis of the datasets

Carter instances, vary in terms of students and examinations sizes, available periods and conflict density. Conflict density is simply computed as $|\mathbb{E}|/|\mathbb{V}|^2$ and represents how likely it is for two examinations to have students in common. Table I presents features of the datasets. We observed that, rather strangely, there are students that are not enrolled in any examination. In particular, in ute-s-92 such student is student 921 and in pur-s-93, such students are students 583, 20075 and 20717. Since, the cost is computed by dividing the total penalty of a solution with the number of students, a small distortion enters the calculation for ute-s-92 and pur-s-93.

TABLE I
Carter Datasets

Dataset	Examinations	Students	Enrollments	Periods	Conflict Density
car-f-92	543	18419	55522	32	0.1377
car-s-91	682	16925	56877	35	0.1282
ear-f-83	190	1125	8109	24	0.2655
hec-s-92	81	2823	10632	18	0.4155
kfu-s-93	461	5349	25113	20	0.0555
lse-f-91	381	2726	10918	18	0.0624
pur-s-93	2419	30032	120681	42	0.0295
rye-s-93	486	11483	45051	23	0.0751
sta-f-83	139	611	5751	13	0.1430
tre-s-92	261	4360	14901	23	0.1800
uta-s-92	622	21266	58979	35	0.1254
ute-s-92	184	2750	11793	10	0.0845
yor-f-83	181	941	6034	21	0.2873

To further analyze the Carter dataset we used Python and NetworkX [11] package which is an excellent tool for working with graphs.

1) Graph coloring: Given a certain number of periods the goal is to schedule each examination to a period, while avoiding positioning to the same period examinations with common students. The NetworkX's `greedy_color(`$\mathbb{G}$`)` function can be used in order to achieve this. Currently, NetworkX supports seven base coloring strategies (Largest First, Random Sequential, Smallest Last, Independent Set, Connected Sequential Breadth First Search, Connected Sequential Depth First Search and Saturation Largest First) while five of them have variations that use color interchange. Color interchange improves upon the efficiency of the base algorithm. When, a new color is needed by the algorithm, it is examined if it is possible to swap colors in a bichromatic subgraph of graph $\mathbb{G}$ and thus avoid the need for the new color [12].

Table II shows for each dataset the coloring algorithms that managed to produce a feasible coloring. For each algorithm, next to its name the ratio of successful colorings achieved is shown. It should be noted that in several occasions the number of colors that was returned was lower than the cutoff value. Figure 1 shows for the three best performing algorithms (SLI, LFI and DS) the exact periods needed for "coloring" each Carter dataset.

Fig. 1. Colors needed by DS, LFI and SLI in Carter datasets

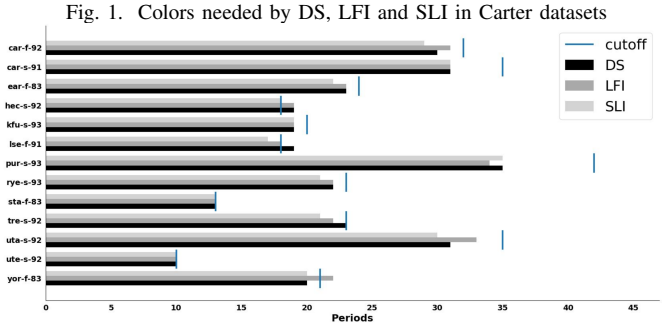


TABLE II
COLORING ALGORITHMS

Dataset	Algorithms that produce valid colorings for $\mathbb{G}$ (100 runs for RS and RSI)
car-f-92	DS, LF, LFI, SLI
car-s-91	DS, LF, LFI, SL, SLI
ear-f-83	DS, LFI, RSI, SL, SLI
hec-s-92	RSI, SLI
kfu-s-93	DS, CSBI, LF, LFI, RSI, SL, SLI
lse-f-91	LFI, RSI, SL, SLI
pur-s-93	DS, LF, LFI, SL, SLI
rye-s-93	DS, CSBI, LFI, RSI, SL, SLI
sta-f-83	DS, CSB, CSBI, CSD, CSDI, IS, LF, LFI, RS, RSI, SL, SLI
tre-s-92	DS, LF, LFI, RSI, SLI
uta-s-92	DS, LFI, SL, SLI
ute-s-92	DS, CSBI, LFI, RSI, SLI
yor-f-83	DS, RSI, SLI

CSB=Connected Sequential BFS (1/13), CSBI=Connected Sequential BFS with Interchange (4/13), CSD=Connected Sequential DFS (1/13), CSDI=Connected Sequential DFS with Interchange (1/13), DS=DSatur (11/13), IS=Independent Set (1/13), LF=Largest First (6/13), LFI=Largest First with Interchange (11/13), RS=Random Sequential (1/13), RSI=Random Sequential with Interchange (8/13), SL=Smallest Last (8/13), SLI=Smallest Last with Interchange (13/13)

TABLE III
CONNECTED COMPONENTS PER DATASET

Dataset	Components of size > 1	Bridges	Nr. bridge components
car-f-92	540, 2	5	8
car-s-91	675, 3	5	11
ear-f-83	190	0	1
hec-s-92	81	0	1
kfu-s-93	435, 5, 2, 2	10	31
lse-f-91	379	1	4
pur-s-93	2407, 4, 2	12	21
rye-s-93	485	0	2
sta-f-83	62, 47, 30	0	3
tre-s-92	260	1	3
uta-s-92	622	1	2
ute-s-92	177, 7	0	2
yor-f-83	181	0	1

TABLE IV
CONNECTED COMPONENTS OF DATASET STA-F-83

Component	Size	Cost	Cost (dec.)
1	30	16002	26.1899
2	47	47250	77.3322
3	62	32695	53.5106
		95947	157.0327

2) Connected components: Since the UETP problem is nicely represented with graphs, an idea is to find if there are more than one connected components in each one of the Carter datasets. Each such connected component would be a subgraph of the corresponding graph. In general, a graph $\mathbb{G}$ consists of n subgraphs such that for $i \in 1..n$ it holds:

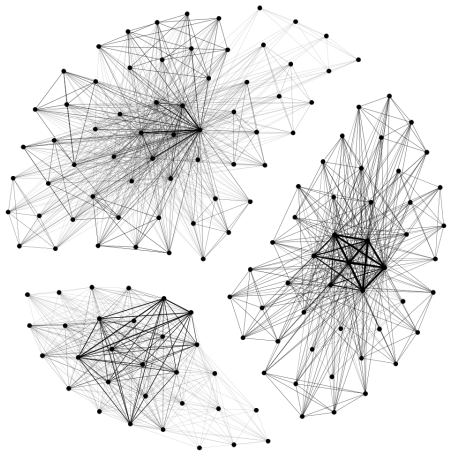
$$\begin{aligned} &SG_i(SV_i, SE_i) \\ &SV_i \subseteq \mathbb{V}, SE_i \subseteq \mathbb{E} \\ &SG_1 \cup SG_2 \cup ... \cup SG_n = \mathbb{G} \\ &SG_1 \cap SG_2 \cap ... \cap SG_n = \emptyset \end{aligned} \tag{1}$$

The existence of independent subgraphs means that the problem can be decomposed and solved separately for each subgraph. We used NetworkX's function `connected_components(`$\mathbb{G}$`)` and found the connected components for all Carter datasets, which are shown in table III. Most of the datasets have connected components but usually there is a big component and a few very small ones. Dataset sta-f-83 is a notable exception since it splits into 3 connected components with 30, 47 and 62 examinations respectively. A visualization of them is shown in figure 2 which shows edges with larger weights (i.e. pairs of examinations with more common students) darker. We tried to solve each component separately using IBM ILOG v.20.1.0 CP Solver and we managed to get the best known result for the dataset analyzed per component as table IV shows. In order to achieve this about 8 hours of solving time per component were needed. Unfortunately, no guarantee that we have reached the optimal solution was given.

Another interesting dataset is ute-s-92, which has two connected components with sizes 177 and 7. The small component can be solved to optimality using IBM ILOG v.20.1.0 CP Solver in a few seconds and returns an objective value of 645 (645/2750=0.2345 in decimal value, where 2750 is the total number of students for ute-s-92). Note, that the cost of this component in the best known solution that is referenced in table VI is also 645.

We also explored the similar idea of finding "bridges" in order to identify decomposition opportunities. A bridge is an edge, once removed reveals a new subgraph. Another NetworkX's function, `bridges(`$\mathbb{G}$`)` was used in order to find bridges in Carter datasets. Then we removed them and counted the number of the resulted connected components. Results are shown at the two rightmost columns of table III.

Fig. 2. Dataset sta-f-83 connected components visualized by Gephi [15]



3) Noise students and noise examinations: Noise students and noise examinations are considered irrelevant to the problem. Some noise examinations can be scheduled to arbitrary periods without affecting the cost of the solution. Additionally,

other noise examinations can be distributed to the available periods inflicting zero contribution to the cost. In [13] a student is marked as noise if he is enrolled in only one examination, while an examination is marked as noise if it is taken only by noise students.

We further expand on the idea of noise examinations by examining subgraph connected components. If a connected component of the graph has size less than $\lfloor \frac{P-1}{6} \rfloor + 1$ then examinations of the connected component can be distributed to the available periods P without any cost. So, these examinations and their corresponding students are in effect noise. In table V individual noise examination numbers are presented, alongside with the number of students that are noise, for every dataset. Noise examinations identified in this ways are shown in bold.

A third way of identifying noise examinations follows. Based on the fact that an examination x scheduled at a period p can introduce penalties with conflicting examinations placed in range $[p-5, p+5]$ we are certain that each examination affects at most 11 periods. Therefore an examination x of degree d_x in a dataset with P available periods where $d_x < \frac{P}{11}$ can always be placed in at least one period without inflicting any cost. By removing examinations under this threshold, recalculating the degrees for the remaining examinations and working recursively until all examinations are above the threshold, we mark the removed examinations as noise. New noise examinations identified only in this way are bold and underlined in table V.

So, now noise students become a) students enrolled in a single examination, b) students that are enrolled in examinations of connected components of $\mathbb{G}$ that inflict no cost and c) students that have at most one non-noise examination and an arbitrary number of noise examinations.

TABLE V
NOISE

Dataset	Noise examinations	Nr. of noise students
car-f-92	**127**, **233**, **253**, **254**, **327**, **328**, **450**, **496**, **518**, 519	3983
car-s-91	**16**, 33, **92**, **331**, **332**, **333**, 349, 440, **441**, **654**, **655**, **656**, 657	3435
ear-f-83	$\emptyset$	1
hec-s-92	$\emptyset$	321
kfu-s-93	6, 16, 22, 50, **70**, 95, **110**, 122, **138**, **139**, 178, 204, 205, 216, **236**, **237**, 285, **310**, ***312***, **313**, **314**, 329, 330, **345**, 355, 369, 381, **412**, 443	288
lse-f-91	168, 256, **376**	100
pur-s-93	**24**, **32**, **136**, 153, **164**, **166**, **167**, **333**, **340**, **341**, **342**, **343**, **344**, **346**, **347**, **415**, **425**, 552, **553**, **554**, **555**, **567**, **569**, **570**, **613**, **614**, **616**, **733**, **737**, **745**, **802**, **896**, **898**, **924**, **936**, **937**, **944**, **945**, **946**, **969**, **970**, **972**, **975**, 976, **980**, **981**, **982**, 983, **1341**, **1413**, **1415**, **1416**, **1435**, **1439**, 1454, 1520, **1665**, **1740**, **1743**, **1751**, **1909**, **1971**, **2012**, **2013**, **2014**, **2127**, **2128**, **2129**, **2130**, **2131**, **2132**, **2133**, **2135**, **2137**, **2138**, **2139**, **2198**, **2337**, **2394**, **2395**, **2396**	2764
rye-s-93	304	2025
sta-f-83	$\emptyset$	0
tre-s-92	**61**, **62**, 186	672
uta-s-92	**233**, **480**, **557**, **560**, **564**	6183
ute-s-92	$\emptyset$	79
yor-f-83	$\emptyset$	1

IV. LOWER BOUNDS

In this section we explore the possibility of finding lower bounds for the Carter datasets. Firstly, we examine the question of finding the maximum possible number of students that can participate, at the same period, in an examination. This question can be answered by solving the following compactly formulated problem which resembles graph coloring and knapsack altogether.

$$\begin{aligned} \min \quad & \sum_{x \in \mathbb{X}} r_x y_x \\ \text{s.t.} \quad & y_{x_1} + y_{x_2} \leq 1, \qquad \forall (x_1, x_2) \in \mathbb{E} \\ & y_x \in \{0,1\}, \quad x \in \mathbb{X} \end{aligned} \tag{2}$$

The number of student enrollments that each examination $x \in \mathbb{X}$ has is given by the parameter value r_x. The y_x decision variable assumes value 1 when examination x is included in the abstract period with the maximum possible number of students and 0 otherwise. The inequality guarantees that no examinations with common students, as denoted by the existence of an edge in the corresponding graph $\mathbb{G}$, should be scheduled at the abstract period.

The problem can be optimally solved for all Carter datasets but one (pur-s-93) and the results are presented at the second column of Table VI. Values in bold denote that optimality was reached. Dataset pur-s-93 reached value 12126 using the IBM CPLEX v.20.1.0 MIP Solver [14], in 24 hours, achieving a gap of 9.52% from the bound of the optimal max students value (13279.92).

Secondly, we revisit the idea of computing lower bounds from [13]. Given that each student will eventually participate in examinations at certain periods among the available ones we can enumerate all possible schedules as patterns of zeros and ones of length P where 1 denotes participation in an examination and 0 means no examination in this period for the student. The cost of each pattern is easily computed and by summing the minimum cost among all possible patterns for each student we get a lower bound for the problem. We can speedup calculations by exploiting the fact that if the number of examinations of a student is less than or equal to $\lfloor \frac{P-1}{6} \rfloor + 1$ then the examinations of the student can be spread with zero cost among the available periods, so no cost computation is necessary. Furthermore, since no difference exists at this point among students with the same number of examinations we can compute penalty once and use the same value many times for all students of each group, grouped by the number of examinations taken. Nevertheless, this approach fails for datasets car-s-91, car-f-92, pur-s-93 and uta-s-92 since the sizes of permutations for those cases are too big. For example, for the case of pur-s-93 since it has 42 periods and students who are enrolled to a maximum of 9 courses, a naive way of computing cost for each pattern would require $\sum_{i=1..9} \binom{42}{i} = 597121090$ cost evaluations. So, another model was formulated. For each student the

examinations that he participates is known. We can relax the explicit examinations for each student, and substitute them with anonymous examinations identified by their order in each student's schedule. For example, a student might be enrolled in three examinations: k, l and m. In our formulation we drop the information about the specific examinations and we define three variables that specify the periods of the first, second and third examination in the student's schedule. It is possible that any of the examinations among k, l or m will be the first, the second and third taken examination, provided that each examination will occur only once. It should be noted that we can still compute the cost for each possible student's schedule, the same way as it is computed for the original problem. So, the problem becomes that of minimizing the total cost by deciding about the periods that each student will participate in examinations. In this formulation decision variables $x_{s,i}$ are integer and assume the value of the period that the i^{th} in order among examinations of student s is scheduled to occur. Note that the values $x_{s,i}$ are in strictly increasing order. The cost of the solution is computed by assigning values to binary variables $ya_{s,i,k}, yb_{s,i,k}, yc_{s,i,k}, yd_{s,i,k}$ and $ye_{s,i}$. Variables $ya_{s,i,k}$ assume value 1 when examination i and examination $i+1$ of student s impose k penalty points and 0 otherwise. Variables $yb_{s,i,k}$ assume value 1 when examination i and examination $i+2$ of student s impose k penalty points and 0 otherwise. Likewise for variables $yc_{s,i,k}, yd_{s,i,k}$ and $ye_{s,i,k}$. The resulted formulation is presented in equations 3.

$$
\begin{aligned}
\text{min} \quad & \sum_{s\in S}\Bigg(\sum_{k\in\{16,8,4,2,1\}}\sum_{i\in 1..|\mathbb{X}_s|-1} ya_{s,i,k}*k \\
& + \sum_{k\in\{8,4,2,1\}}\sum_{i\in 1..|\mathbb{X}_s|-2} yb_{s,i,k}*k \\
& + \sum_{k\in\{4,2,1\}}\sum_{i\in 1..|\mathbb{X}_s|-3} yc_{s,i,k}*k \\
& + \sum_{k\in\{2,1\}}\sum_{i\in 1..|\mathbb{X}_s|-4} yd_{s,i,k}*k + \sum_{i\in 1..|\mathbb{X}_s|-5} ye_{s,i}\Bigg) \\
\text{s.t.} \quad & x_{si}+1 \le x_{si+1}, \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-1 \\
& ya_{s,i,16} = (x_{s,i+1}-x_{s,i}==1), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-1 \\
& ya_{s,i,8} = (x_{s,i+1}-x_{s,i}==2), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-1 \\
& ya_{s,i,4} = (x_{s,i+1}-x_{s,i}==3), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-1 \\
& ya_{s,i,2} = (x_{s,i+1}-x_{s,i}==4), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-1 \\
& ya_{s,i,1} = (x_{s,i+1}-x_{s,i}==5), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-1 \\
& yb_{s,i,8} = (x_{s,i+2}-x_{s,i}==2), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-2 \\
& yb_{s,i,4} = (x_{s,i+2}-x_{s,i}==3), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-2 \\
& yb_{s,i,2} = (x_{s,i+2}-x_{s,i}==4), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-2 \\
& yb_{s,i,1} = (x_{s,i+2}-x_{s,i}==5), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-2 \\
& yc_{s,i,4} = (x_{s,i+3}-x_{s,i}==3), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-3 \\
& yc_{s,i,2} = (x_{s,i+3}-x_{s,i}==4), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-3 \\
& yc_{s,i,1} = (x_{s,i+3}-x_{s,i}==5), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-3 \\
& yd_{s,i,2} = (x_{s,i+4}-x_{s,i}==4), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-4 \\
& yd_{s,i,1} = (x_{s,i+4}-x_{s,i}==5), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-4 \\
& ye_{s,i} = (x_{si+5}-x_{si}==5), \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-5 \\
& x_{s,i}\in 1..P \\
& ya_{s,i,k}\in\{0,1\} \quad \forall s\in S, k\in\{16,8,4,2,1\}, i\in 1..|\mathbb{X}_s|-1 \\
& yb_{s,i,k}\in\{0,1\} \quad \forall s\in S, k\in\{8,4,2,1\}, i\in 1..|\mathbb{X}_s|-2 \\
& yc_{s,i,k}\in\{0,1\} \quad \forall s\in S, k\in\{4,2,1\}, i\in 1..|\mathbb{X}_s|-3 \\
& yd_{s,i,k}\in\{0,1\} \quad \forall s\in S, k\in\{2,1\}, i\in 1..|\mathbb{X}_s|-4 \\
& ye_{s,i}\in\{0,1\} \quad \forall s\in S, i\in 1..|\mathbb{X}_s|-5
\end{aligned}
\tag{3}
$$

The above formulation is able to find lower bounds for all datasets, including the "big" ones, provided that the model is implemented in a capable MIP solver like IBM CPLEX v.20.1.0 which we used. Note that components of equations of the form $a == b$ assume value 1 when expression a equals expression b and 0 otherwise. Column "LB I" of Table VI shows the computed values. A further improvement on some lower bounds is possible by exploiting the information about the theoretical maximum number of students than can possibly be in a period (maxs). This is done by adding the following constraint to the model.

$$\sum_{s\in S}\sum_{i\in 1..|\mathbb{X}_s|}(x_{s,i}==p) \le maxs, \quad \forall p\in 1..P \tag{4}$$

The bounds that use the maximum possible number of students are shown in column "LB II". Again bold values indicate that optimality was reached in computing the bound by the solver. For the case of pur-s-93, since we don't have the optimal value for the maximum number of students that can take examinations in a single period, the bound is not computed. Column "LB II (dec.)" shows the value of column "LB II" divided by the number of students of the respective dataset, which is the standard way of evaluating a solution's cost in Carter datasets. The last column shows the best known solutions from [8].

TABLE VI
MAXIMUM POSSIBLE NUMBER OF STUDENTS IN A SINGLE PERIOD, LOWER BOUNDS (LBS) AND BEST KNOWN SOLUTION VALUES

Dataset	Max students	LB I	LB II	LB II (dec.)	Best known
car-f-92	**4571**	**145**	145	0.0079	3.6421
car-s-91	**4707**	**100**	100	0.0059	4.2379
ear-f-83	**870**	**20078**	**20542**	18.2596	32.4204
hec-s-92	**1382**	**9865**	**10773**	3.8162	10.0337
kfu-s-93	**3906**	**30136**	**30682**	5.7360	12.7990
lse-f-91	**1611**	**7537**	**9147**	3.3555	9.7737
pur-s-93	12126	**42**	N/A	0.0014	4.0005
rye-s-93	**7108**	**43484**	43484	3.7868	7.8376
sta-f-83	**611**	**92900**	**92900**	152.0458	157.0327
tre-s-92	**1461**	**2588**	**3750**	0.8601	7.5904
uta-s-92	**4856**	**46**	46	0.0022	2.9472
ute-s-92	**2432**	**59152**	**59398**	21.5993	24.7600
yor-f-83	**698**	**17842**	**18014**	19.1435	34.4049

Finally, we conclude that lower bounds can indeed be computed, but they are close to actually achieved solutions only for cases where the number of periods is rather small and students are enrolled to many courses. For other cases they are computationally hard to compute and too loose to be of practical importance.

V. SYMMETRIES

In this section we explore the very structure of the problem and identify symmetries. We propose ways of exploiting them resulting, hopefully, in a domain space less difficult to be searched. Exact and approximate solvers are expected to benefit from symmetry breaking.

A. Bidirectional symmetry

It can be observed that if the period p_x that each examination x is scheduled to, changes to $P - p_x$, then we effectively get the original solution reversed. Since the cost is computed based on the distance among periods of scheduled examinations, it stays unaffected when reverse occurs, as demonstrated in equation 5.

$$\left|p_{x_i} - p_{x_j}\right| = \left|P - p_{x_i} - (P - p_{x_j})\right| \quad \forall x_i, x_j \in \mathbb{X} \tag{5}$$

So, in order to break this type of symmetry, two examinations x_i and x_j with common students can be selected and impose inequality 6.

$$p_{x_i} < p_{x_j} \quad \forall x_i, x_j \in \mathbb{X} : (x_i, x_j) \in \mathbb{E} \tag{6}$$

B. Interchangeable examinations

Let $\mathbb{I}$ be a set of sets where all examinations in a set are not adjacent with other examinations in the same set and each examination has edges to the same other examinations with equal weight values. In mathematical terms examinations x_i and x_j with neighbor sets b_{x_i} and b_{x_j} respectively belong to set $\mathbb{I}$ if equations in 7 hold. We argue that all examinations in each set of $\mathbb{I}$ must be scheduled at the same "best" period in an optimal solution. Since the examinations of each set are not in conflict, and each examination conflicts with exactly the same other examinations, there is no benefit in positioning them in different periods. An example of the symmetry is shown in figure 3 involving examinations 3 and 4.

$$\begin{aligned} &b_{x_i} \equiv b_{x_j} \\ &w_{x_i,x_n} = w_{x_j,x_n} \quad \forall x_n \in b_{x_i} \end{aligned} \tag{7}$$

The symmetry involving examinations in $\mathbb{I}$ can be broken by imposing constraints that schedule at the same period, all examinations of each set in it.

Another symmetry is revealed by defining set $\mathbb{I}^+$ as a set of sets where all examinations in a set are adjacent with all other examinations in the same set and each examination has edges to the same other examinations using the same weight values. The set $\mathbb{I}^+$ is formally defined in equations 8, where the symmetric difference of neighbor sets b_{x_i} and b_{x_j} of every pair of examinations x_i and x_j in a set of $\mathbb{I}^+$ is equal to $\{x_i, x_j\}$. An example of the symmetry is shown in figure 3 involving examinations 1 and 2.

$$\begin{aligned} &b_{x_i} \triangle b_{x_j} = \{x_i, x_j\} \\ &w_{x_i,x_n} = w_{x_j,x_n} \quad \forall x_n \in b_{x_i} : x_n \neq x_j \end{aligned} \tag{8}$$

The symmetry involving examinations in $\mathbb{I}^+$ can be broken by imposing constraints to artificially order (e.g. by id) examinations belonging to the same set.

Table VII contains sizes of identified sets for $\mathbb{I}$ and $\mathbb{I}^+$ for the Carter datasets.

Expanding on the idea of sets $\mathbb{I}$ and $\mathbb{I}^+$ we can identify pairs of examinations that have the same set of neighbors but perhaps different weights on their edges. These pairs of examinations can swap their periods without violating hard constraints but may effect the objective. As hard constraints are not violated, this exchange can be used as a neighborhood operator for exploring the feasible space in the same way that other moves that result in feasible schedules, like Kempe chain [16] moves and swap periods moves, do.

Fig. 3. Examinations 3,4 $\in \mathbb{I}$ and examinations 1,2 $\in \mathbb{I}^+$

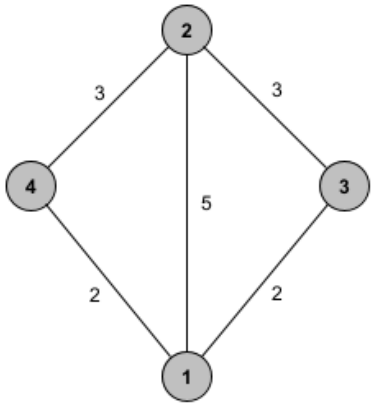


TABLE VII
INTERCHANGEABLE EXAMINATIONS OF TYPE $\mathbb{I}$ AND TYPE $\mathbb{I}^+$

Dataset	Set sizes in $\mathbb{I}$	Set sizes in $\mathbb{I}^+$
car-f-92	-	2
car-s-91	4	-
ear-f-83	-	5, 2, 2
hec-s-92	-	-
kfu-s-93	17, 2	2, 2, 2, 2, 2, 2, 2, 2
lse-f-91	2, 2	5, 3, 2, 2, 2, 2, 2, 2, 2
pur-s-93	6	2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
rye-s-93	-	4, 3, 3, 3, 2, 2
sta-f-83	-	5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5
tre-s-92	-	-
uta-s-92	-	-
ute-s-92	-	2, 2, 2
yor-f-83	-	-

VI. MATHEMATICAL FORMULATIONS FOR "BIG" MOVES

Two mathematical models are presented in this section. The first one deals with finding initial feasible solutions and then optimizing them. The second one, starts with a feasible solution, and permutes periods without letting examinations that are already scheduled at the same period to end up in different periods regarding each other.

A. Exact model

In this section we present a simple model that defines an integer decision variable p_{x_i} for every examination x_i, which assumes the number of the period that the examination will be scheduled to. The penalty that each pair of examinations x_i and x_j inflicts upon the solution is based on the periods that the examinations are scheduled to, and is computed based on equation 9. The formulation of the problem is presented in equations 10.

$$f_{x_i,x_j} = \begin{cases} 2^{5-|p_{x_i}-p_{x_j}|}, & \text{if } |p_{x_i}-p_{x_j}| \leq 5 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

$$\begin{aligned} \min \quad & \sum_{(x_i,x_j)\in\mathbb{E}} w_{x_i,x_j} * f_{x_i,x_j} \\ \text{s.t.} \quad & p_{x_i} \neq p_{x_j} \quad \forall (x_i,x_j) \in \mathbb{E} \\ & p_{x_i} \in 1 \dots P \quad \forall x_i \in \mathbb{X} \end{aligned} \quad (10)$$

Unsurprisingly, the above model fails to return good enough solutions for Carter datasets, due to their sizes. Nevertheless, by replacing the objective function with a dummy function or a function that computes only a part of the objective function (e.g. penalize only consecutive examinations), promising solutions are found. Moreover, by providing an initial solution to the problem, and enforcing that parts of the solution (e.g. specific examinations or examinations in entire periods) would be fixed and only allow changes in the remaining examinations the model can serve as a "large neighborhood" move.

B. Permuting periods model

In this section we present another model that permutes periods filled with examinations in an effort to achieve new solutions with better cost. The decision variables of the model are binary variables $z_{i,j}$ that assume value 1 when all examinations in period i are moved to period j and 0 otherwise. Since we start from a feasible solution, we can compute the common students that exist among two periods i and j and keep the value in parameter $c_{i,j}$. Finally, auxiliary binary variables $y_{i,j,k,l}$ are defined according to equation 11 and indicate the situation of moving examinations in periods i and j to positions k and l respectively. The model is presented in equations 12.

$$y_{i,j,k,l} = \begin{cases} 1, & \text{if } z_{i,k} = z_{j,l} = 1 \text{ and } |k-l| \in [1,5] \\ 0, & otherwise \end{cases} \quad (11)$$

$$\begin{aligned} \min \quad & \sum_{i\in 1\dots P} \sum_{j\in i+1\dots P} \sum_{k\in 1\dots P} \sum_{\substack{l\in 1\dots P: \\ |k-l|\leq 5}} 2^{5-|k-l|} * c_{i,j} * y_{i,j,k,l} \\ \text{s.t.} \quad & \sum_{i\in 1\dots P} z_{i,k} = 1 \quad \forall k \in 1\dots P \\ & \sum_{k\in 1\dots P} z_{i,k} = 1 \quad \forall i \in 1\dots P \\ & z_{i,k} \in \{0,1\} \quad \forall i,k \in 1\dots P \end{aligned} \quad (12)$$

The model can lower the cost of initial feasible solutions and act as another "large neighborhood" move.

VII. Conclusion

In this paper we approached the UETP variation of the examination timetabling problem and hopefully presented insights about the problem that were hidden in spite of the great research effort that has been undertaken during the last 25 years. We analyzed the benchmark Carter datasets, found connected components for some of the problems, and opportunities for decomposition. We have found more noise students and noise examinations, which suggests that the benchmark problems can be substituted by "core" versions of them. We also found new, better bounds for some benchmark problems and identified symmetries that exist in the problem and hardens search approaches and exact solvers. Finally, we presented two mathematical formulations for UETP that can serve as large neighborhood moves.

We plan to use our findings in order to contribute high quality solutions to the community. We expect that systematic search of the domain space using traditional moves (swap examinations, Kempe chains, ejection chains, etc) and "large neighborhood" moves, backed up by exact solvers (MIP, CP, and others) has great potential in finding solutions close to the best known ones.

References

[1] M. W. Carter, G. Laporte, S. Y. Lee "Examination Timetabling: Algorithmic Strategies and Applications". Journal of the Operational Research Society, vol. 47, pp. 373–383, 1996.

[2] R. Qu, E. K. Burke, B. McCollum, L. T. Merlot, S. Y. Lee "A survey of search methodologies and automated system development for examination timetabling". Journal of scheduling, 12(1), pp. 55–89, 2009

[3] B. Aldeeb, M. A. Al-Betar, A. O. Abdelmajeed, M. J. Younes, M. AlKenani, W. Alomoush "A Comprehensive Review of Uncapacitated University Examination Timetabling Problem". International Journal of Applied Engineering Research, vol. 14, pp. 4524-4547, 2019.

[4] E. K. Burke, D. G. Elliman, R. Weare "A university timetabling system based on graph colouring and constraint manipulation". Journal of research on computing in education, 27(1), pp 1–18, 1994.

[5] M. Caramia, P. Dell'Olmo, G. F. Italiano "Novel Local-Search-Based Approaches to University Examination Timetabling". INFORMS Journal on Computing, vol. 20, pp. 86–99, 2008.

[6] N. Leite, C. M. Fernandes, F. Melicio, A. C. Rosa "A cellular memetic algorithm for the examination timetabling problem". Computers & Operations Research, vol. 94, pp. 373–383, 2018.

[7] N. F. Jamaluddin, N. A. H. Aizam "Timetabling communities' demands for an effective examination timetabling using integer linear programming". International Journal of Mathematical and Computational Sciences, 10(5), pp. 263–268, 2016

[8] R. Bellio, S. Ceschia, L. Di Gaspero, A. Schaerf "Two-stage multi-neighborhood simulated annealing for uncapacitated examination timetabling". Computers & Operations Research, 132, 105300, 2021.

[9] R. Qu, F. He, E. K. Burke "Hybridizing integer programming models with an adaptive decomposition approach for exam timetabling problems". Proceedings of the 4th Multidisciplinary International Scheduling: Theory and Applications, pp. 435–446, 2009.

[10] H. Turabieh, S. Abdullah "An integrated hybrid approach to the examination timetabling problem". Omega, 39(6),pp. 598–607, 2011.

[11] A. A. Hagberg, D. A. Schult, P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX". Proceedings of the 7th Python in Science Conference (SciPy2008). Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11—15, 2008.

[12] M. M. Sysło, M. Deo, J. S. Kowalik, Discrete Optimization Algorithms with Pascal Programs, pp. 415-424, 1983.

[13] P. Alefragis, C. Gogos, C. Valouxis, E. Housos "A multiple metaheuristic variable neighborhood search framework for the Uncapacitated Examination Timetabling Problem". Proceedings of the 13th International Conference on the Practice and Theory of Automated Timetabling-PATAT, vol. 1, pp. 159–171, 2021.

[14] Cplex V20.1.0: User's Manual for CPLEX, IBM ILOG, International Business Machines Corporation, 2021.

[15] M. Bastian, S. Heymann, M. Jacomy "Gephi: an open source software for exploring and manipulating networks". International AAAI Conference on Weblogs and Social Media, 2009.

[16] A. B. Kempe, "On the geographical problem of the four colours", American J. of Math., 2(3),193-200, 1879.