A proven optimal result for a benchmark instance of the Uncapacitated Examination Timetabling Problem

Angelos Dimitsas[1], Vasileios Nastos[1], Christos Valouxis[2], Panayiotis Alefragis[3], and Christos Gogos[1]

[1] Dept. of Informatics and Telecommunications University of Ioannina, Arta, Greece
{a.dimitsas,vnastos,cgogos}@uoi.gr
[2] Dept. of Electrical and Computer Engineering, University of Patras, Greece
cvalouxis@upatras.gr
[3] Dept. of Electrical and Computer Engineering, University of Peloponnese,Patras, Greece alefrag@uop.gr

Abstract. Examination timetabling is a problem well known to the scheduling community. Its simplest version, which is the Uncapacitated Examination Timetabling Problem is easily described and comprehended. Nevertheless, proof of optimality is notoriously difficult even for moderate size problems. In this paper we describe the effort that our team exercised in finally proving the optimality of the sta83 instance of Carter's dataset. The problem was decomposed naturally in three parts and for each part a different approach managed to prove optimality of the currently best known solution. Several hours of computation were needed, but now we are confident that no solution exists with cost less than the proved optimal value. This work also presents optimal solutions to subproblems that exist in various public datasets problems and two best known solutions of such problems.

Keywords: Examination Timetabling · Mixed Integer Programming · Heuristics

1 Introduction

Timetabling problems arise in several domains including health-care, education, call centers, airlines and others. Rostering and scheduling are also commonly used terms to describe timetabling problems. In this paper we study the Uncapacitated Examination Timetabling Problem (UETP). UETP is the problem of scheduling university examinations to periods (time-slots) in such a way that no student should be examined at the same period for more than one course. Furthermore, the schedule of each student should allow enough time for studying between successive examinations. The problem is uncapacitated in the sense that no room capacities or availabilities are considered.

Our contribution to UETP is twofold. Firstly, we present a way of decomposing and reducing the sizes of the problems that results in obtaining two new best known solutions for benchmark instances. Secondly, and most noteworthy we propose a novel way of approaching a certain known instance of the Carter's dataset [6] of the UETP that results in actually proving the optimal value of the instance.

An outline of the paper follows. Section 2 provides a succinct description of the problem. Section 3 presents a glimpse of the broad bibliography for university examination problems capacitated or not. The next section describes our efforts to cleanse and decompose the problem instances so as to reduce their sizes, in an effort to feed various solution approaches with easier to digest problems. Section 5 is devoted to attacking the UETP problem instances with three specific methods that are later used in Section 6 to prove optimality for problem instance sta83 of the well known Carter's dataset. Next, our conclusions follow.

2 Problem Description

Each UETP instance contains information about the set of examinations that each student is enrolled in. Each instance has a specific number of periods that can be used to schedule the examinations to. The single hard constraint is that no student is allowed to participate in more than one examination per period. To allow time for each student to study between his examinations, for each student s, for each pair of examinations taken by s, a penalty of 16 is imposed if the two examinations occur in adjacent time slots (called distance 1), penalty 8 is imposed for distance 2, 4 for distance 3, 2 for distance 4, and 1 for distance 5.

The natural way to represent an instance is as an undirected weighted graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ where each vertex in $\mathbb{V}$ is an examination and each edge in $\mathbb{E}$ connects two examinations with common students. The weight of each edge is the number of common students for the examinations it connects.

3 Related work

The field of educational timetabling is very active. Several papers are typically published every year regarding course timetabling (post-enrollment and curriculum-based), examination timetabling, high school timetabling, thesis defense timetabling and others. Several surveys regarding the field have been published and present the challenges that such problems pose [12], [9]. The survey by Qu et al. [10] focuses on examination timetabling that is the subject of our work too. Recent surveys by Tan et al. [13] and Ceschia et al. [7] demonstrate the strong interest of the timetabling community for educational timetabling problems. Maybe this can be justified by the familiarity of such problems to academia circles. In [7] focus is given on "standard" formulations and benchmark instances that are also used in our work are presented. Another recent work for real world examination timetabling problems, this time, is the paper from Battistutta et al. [3].

Our team is also active in educational timetabling. In [8] we proposed a novel way of estimating lower bounds for UETP instances. Ideas about symmetry elimination, problem decomposition and cleansing of the instances were also presented there. The current paper serves as a follow-up and provides experimental results based on those ideas.

4 Preprocessing

Before solving each problem we perform a cleansing process through which we remove problem components that are insignificant and only add noise to the problem. The premise is that by solving the cleansed problem, we will still be able to find optimal solutions that will be optimal for the original problem. Furthermore, we identify independent subproblems that exist in each problem. Such subproblems can be solved independently and the solution to the original problem can be stitched by using the solutions of the subproblems. An exploration of the main ideas that we use for cleansing and decomposing the problems are more extensively described in our latest paper [8]. A synopsis follows.

Initially, we remove obvious noise students and examinations [2] (see lines 1 and 2 of Algorithm 1). Then, we identify subgraphs of the graph that can be handled independently. Note that the size of a subgraph refers to the number of its nodes which is equal to the number of the corresponding subproblem's exams. Subgraphs of size lower than $\lfloor \frac{P-1}{6} \rfloor + 1$ are identified as noise. This can be justified by the fact that we can spread the examinations of such subgraphs to the P available periods with zero penalty. Examinations with degree lower than $\frac{P}{11}$ are also noise since they can be always positioned with zero penalty. Then, any student that has a single non-noise examination and an arbitrary amount of noise examinations is also considered as noise. The process repeats until no more examinations or students can be marked as noise. A description of the procedure is given in Algorithm 1.

Another form of preprocessing involves the identification of interchangeable examinations that was proposed in [8]. These examinations have the same neighborhoods, as defined in graph $\mathbb{G}$, and the same number of common students for each neighbor. As these examinations are practically the same we can enforce them to either be in the same period if they are not in conflict or to follow a specific sequence of appearance in the final schedule if they have common students. By eliminating this type of symmetry of the problem, MIP/CP solvers are able to better explore the solution space.

4.1 Datasets

The standard benchmark dataset for UETP is Carter's dataset (a.k.a. Toronto dataset). Those instances were contributed in [6] back in 1996 and since then were used in many papers. Recently, 19 new instances that are modified versions of other more complex formulations, were added by Bellio et al. [4]. All of them are publicly available in https://opthub.uniud.it/ which is a site that

Algorithm 1: Remove noise examinations and students from an examination timetabling problem

```
Input: An examination timetabling problem represented as a graph G
Output: Graph G with noise examinations and noise students removed
Find students enrolled in a single exam, tag them as noise
Find examinations with only noise students, tag them as noise
Remove tagged examinations and students from G
do                // loops until no more noise examinations are found
    done = True
    Let S be the set of disconnected components (subgraphs) of G
    while S ≠ ∅ do
        G_i = next(S)          // i is the identifier of the subgraph
        if |G_i| < ⌊(P−1)/6⌋ + 1 then
            Tag all examinations of G_i as noise
            Tag all students enrolled in examinations of G_i as noise
            Remove tagged examinations and students from G_i and G
            done = False
        do
            more_noise = False
            for each examination e in G_i do
                if deg_e < P/11 then
                    Tag e as a noise
                    Tag all students enrolled in e as noise
                    Remove tagged examinations and students from G_i and G
                    more_noise = True
                    done = False
        while more_noise
        Remove G_i from S
while not done
```

hosts definitions, datasets and solutions of several timetabling problems that have attracted the interest of the timetabling community.

The characteristics of the instances used in this paper are shown in Table 1. Conflict density is a metric that is computed by dividing the number of edges of the problem's corresponding graph by $n(n-1)/2$, where n is the number of vertices. Values for noise students and examinations are computed based on Algorithm 1. Moreover, the table presents the best known values that were obtained by solutions that we have downloaded from https://opthub.uniud.it/ in April 2022. Costs assume integer values and since the problem is of minimization nature, lower values are favored. Normalized costs are shown in the rightmost column of the table and are computed by dividing each integer cost by the corresponding number of students. The star symbol ($*$) in best known cost (95947) of instance sta83 indicates that this cost is optimal. At Section 6 we show that this is indeed the case. We consider it as the highlight of our work, since it is the first instance among the Carter's dataset for which it is proven that an optimal solution has been reached. It should be noted that the table has symbol † for the best known costs of two instances, ITC2007_9 and ITC2007_10. These best known values were contributed by our team and were obtained by exploiting the concept of noise examinations and students and the decomposition of problems to subproblems that enabled us to use optimal solutions to independent subproblems and search for good solutions using the Variable Neighborhood Search approach described in [2].

Table 1. Instances - descriptive statistics - noise examinations and noise students - best known costs

Instance id	Exams	Students	Periods	Conflict density	Noise exams	Noise students	Best known cost	Best known normalized cost
car92	543	18419	32	0.137986	10	3969	67084	3.6421
car91	682	16925	35	0.128386	13	3409	71727	4.2379
ear83	190	1125	24	0.266945	0	1	36473	32.4204
hec92	81	2823	18	0.420679	0	321	28325	10.0337
kfu93	461	5349	20	0.055579	33	276	68462	12.7990
lse91	381	2726	18	0.062592	3	99	26643	9.7737
pur93	2419	30029	42	0.029495	83	2627	120144	4.0009
rye93	486	11483	23	0.075279	1	2025	89999	7.8376
sta83	139	611	13	0.143989	0	0	*95947	*157.0327
tre92	261	4360	23	0.180696	3	667	33094	7.5904
uta92	622	21266	35	0.125557	5	6180	62675	2.9472
ute92	184	2749	10	0.084937	0	78	68090	24.7690
yor83	181	941	21	0.288889	0	1	32375	34.4049
ITC2007_1	607	7883	54	0.050495	25	227	5628	0.7139
ITC2007_2	870	12484	40	0.011695	238	2430	1538	0.1232
ITC2007_3	934	16365	36	0.026187	124	1306	20768	1.2690
ITC2007_4	273	4421	21	0.149968	0	4	47869	10.8276
ITC2007_5	1018	8719	42	0.008693	343	407	1567	0.1797
ITC2007_6	242	7909	16	0.061555	15	2622	30343	3.8365
ITC2007_7	1096	13795	80	0.019323	358	2620	262	0.0190
ITC2007_8	598	7718	80	0.045489	101	229	409	0.0530
ITC2007_9	169	624	25	0.078402	26	9	†2909	†4.6619
ITC2007_10	214	1415	32	0.049713	53	91	†12184	†8.6106
ITC2007_11	934	16365	26	0.026187	93	1306	54347	3.3209
ITC2007_12	78	1653	12	0.184482	4	684	10631	6.4313
D1-2-17	281	37	38	0.053254	21	1	2428	65.6216
D5-1-17	277	43	45	0.087166	53	0	3653	84.9535
D5-1-18	306	49	45	0.066560	47	0	3245	66.2245
D5-2-17	344	43	45	0.092447	2	0	8362	194.4651
D5-2-18	425	47	59	0.083629	6	0	6619	140.8298
D5-3-18	132	43	22	0.081309	2	0	1406	32.6977
D6-1-18	511	57	60	0.059975	74	0	9793	171.8070
D6-2-18	539	57	78	0.067639	10	0	7883	138.2982

4.2 Decomposed instances

After applying Algorithm 1 some problems are decomposed to subproblems. For most instances a number of examinations and students are removed since they are in effect noise. The resulting subproblems are presented in Table 2. The name of each subproblem follows the pattern d_i_(Ex_Sy_IDz), where d is the name of the originating instance, i is a number that assumes value 1 for the smallest subproblem and is incremented by 1 for each subsequent subproblem (subproblems are ordered by size = number of exams), x is the number of examinations, y is the number of students and z is the smallest examination number that exists in the subproblem. Number z is needed in order to differentiate among subproblems having the same number of examinations and same number of students. This is indeed the case for subproblems D1-2-17_1 and D1-2-17_2 that both have 8 examinations and 1 student but in the first case the identifying examination is 217 while for the second case the identifying examination is 257. Note that in Table 2 the number of examinations and the number of students exclude noise examinations and noise students respectively. Again, the presence of symbol $*$ denotes that the corresponding integer cost is optimal. It should be also noted that the normalized cost is computed by dividing the integer cost by the number of students (including noise ones) that exists in the originating instance.

5 Optimality proving tools

We have identified three different approaches to prove optimality for certain instances, and we present them below. Under certain conditions (number of exams, conflict density, current best known solution, number of periods) these approaches may be able to prove that a solution is indeed optimal.

5.1 Mixed Integer Programming

As optimality is our main concern the first thoughts that come to mind are Linear Programming and Mixed Integer Programming. The mathematical model described below can solve an UETP instance, provided that the instance size is manageable. For a graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ where vertices $\mathbb{V}$ serve as the exams, each edge in $\mathbb{E}$ means that two examinations have common students. The weight of an edge W_{v_1,v_2} connecting vertices v_1 and v_2 is equal to the number of common students these examinations have. P is the number of available periods.

The integer decision variables v_n in Equation 1 denote the period an examination will take place while the derived binary decision variables in Equation 2 help us to activate or deactivate penalties in the objective function in Equation 3. The constraint in Equation 4 forces examinations with common students to take place in different periods. Equation 5 forces binary decision variables in Equation 2 to indicate the distance between two exams. This constraint is not linear but capable solvers like IBM ILOG CPLEX using mathematical modeling tricks are able to linearize it out of the box. Equation 6 allows only one of the penalty indicating variables in Equation 2 to be active at any time. This constraint is redundant but its presence seems to help the solver in reaching better solutions.

$$v_n \in [0, P) \quad \forall n \in 1 \dots |\mathbb{V}| \tag{1}$$

$$\begin{aligned} y16_{v_1,v_2} &\in \{0,1\} \quad \forall (v_1, v_2) \in \mathbb{E} \\ y8_{v_1,v_2} &\in \{0,1\} \quad \forall (v_1, v_2) \in \mathbb{E} \\ y4_{v_1,v_2} &\in \{0,1\} \quad \forall (v_1, v_2) \in \mathbb{E} \\ y2_{v_1,v_2} &\in \{0,1\} \quad \forall (v_1, v_2) \in \mathbb{E} \\ y1_{v_1,v_2} &\in \{0,1\} \quad \forall (v_1, v_2) \in \mathbb{E} \end{aligned} \tag{2}$$

Table 2. Problems resulted by decomposed instances of Table 1 - All noise examinations and noise students are removed

Instance id	Exams	Students	Conflict Density	Best known cost	Best known normalized cost
car92_1(E533_S18328_ID1)	533	18328	0.143139	67084	3.6421
car91_1(E669_S16750_ID1)	669	16750	0.133375	71727	4.2379
ear83_1(E190_S1125_ID1)	190	1125	0.266945	36473	32.4204
hec92_1(E81_S2823_ID1)	81	2823	0.420679	28325	10.0337
kfu93_1(E428_S5194_ID1)	428	5194	0.064326	68462	12.7990
lse91_1(E378_S2724_ID1)	378	2724	0.063576	26643	9.7737
pur93_1(E2336_S29766_ID1)	2336	29766	0.031566	120144	4.0009
rye93_1(E485_S11425_ID1)	485	11425	0.075590	89999	7.8376
sta83_1(E30_S162_ID1)	30	162	0.717241	*16002	*26.1899
sta83_2(E47_S210_ID3)	47	210	0.351526	*47250	*77.3322
sta83_3(E62_S239_ID4)	62	239	0.364357	*32695	*53.5106
tre92_1(E258_S4355_ID1)	258	4355	0.184840	33094	7.5904
uta92_1(E617_S21264_ID1)	617	21264	0.127539	62675	2.9472
ute92_1(E7_S20_ID30)	7	20	0.904762	*645	*0.2346
ute92_2(E177_S2729_ID1)	177	2729	0.090588	67445	24.5344
yor83_1(E181_S941_ID1)	181	941	0.288889	32375	34.4049
ITC2007_1_1(E582_S7798_ID1)	582	7798	0.054563	5628	0.7139
ITC2007_2_1(E9_S33_ID396)	9	33	0.888889	*0	*0.0000
ITC2007_2_2(E623_S9636_ID1)	623	9636	0.020856	1538	0.1232
ITC2007_3_1(E810_S15726_ID1)	810	15726	0.034214	20768	1.2690
ITC2007_4_1(E273_S4421_ID1)	273	4421	0.149968	47869	10.8276
ITC2007_5_1(E11_S9_ID434)	11	9	0.690909	*0	*0.0000
ITC2007_5_2(E13_S41_ID206)	13	41	0.487179	*0	*0.0000
ITC2007_5_3(E14_S263_ID120)	14	263	0.989011	189	0.0217
ITC2007_5_4(E637_S7559_ID1)	637	7559	0.018236	1378	0.1580
ITC2007_6_1(E4_S12_ID5)	4	12	1.000000	*33	*0.0042
ITC2007_6_2(E7_S75_ID122)	7	75	0.666667	*7	*0.0009
ITC2007_6_3(E27_S210_ID9)	27	210	0.293447	146	0.0185
ITC2007_6_4(E189_S7386_ID3)	189	7386	0.093662	30157	3.8130
ITC2007_7_1(E18_S143_ID178)	18	143	0.732026	*0	*0.0000
ITC2007_7_2(E720_S10034_ID2)	720	10034	0.040604	262	0.0190
ITC2007_8_1(E497_S7388_ID1)	497	7388	0.062764	409	0.0530
ITC2007_9_1(E143_S603_ID2)	143	603	0.105683	2909	4.6619
ITC2007_10_1(E7_S81_ID1)	7	81	1.000000	*196	*0.1385
ITC2007_10_2(E9_S91_ID78)	9	91	0.888889	*14	*0.0099
ITC2007_10_3(E11_S29_ID87)	11	29	1.000000	*54	*0.0382
ITC2007_10_4(E12_S111_ID121)	12	111	0.984848	1021	0.7216
ITC2007_10_5(E15_S59_ID200)	15	59	0.857143	292	0.2064
ITC2007_10_6(E16_S220_ID133)	16	220	0.958333	878	0.6205
ITC2007_10_7(E16_S124_ID166)	16	124	0.800000	338	0.2389
ITC2007_10_8(E16_S56_ID51)	16	56	0.550000	76	0.0537
ITC2007_10_9(E17_S143_ID149)	17	143	0.757353	836	0.5908
ITC2007_10_10(E19_S208_ID13)	19	208	0.964912	2356	1.6650
ITC2007_10_11(E23_S215_ID98)	23	215	0.909091	6123	4.3272
ITC2007_11_1(E841_S15857_ID1)	841	15857	0.031989	54347	3.3209
ITC2007_12_1(E5_S62_ID35)	5	62	0.900000	*22	*0.0133
ITC2007_12_2(E69_S1464_ID1)	69	1464	0.232310	10609	6.4180
D1-2-17_1(E8_S1_ID217)	8	1	1.000000	*5	*0.1351
D1-2-17_2(E8_S1_ID257)	8	1	1.000000	*5	*0.1351
D1-2-17_3(E10_S1_ID119)	10	1	1.000000	*17	*0.4595
D1-2-17_4(E11_S1_ID218)	11	1	1.000000	*26	*0.7027
D1-2-17_5(E12_S1_ID189)	12	1	1.000000	*36	*0.9730
D1-2-17_6(E13_S2_ID100)	13	2	0.538462	*0	*0.0000
D1-2-17_7(E14_S1_ID173)	14	1	1.000000	*62	*1.6757
D1-2-17_8(E18_S1_ID1)	18	1	1.000000	*150	*4.0541
D1-2-17_9(E18_S1_ID51)	18	1	1.000000	*150	*4.0541
D1-2-17_10(E28_S2_ID7)	28	2	0.592593	*190	*5.1351
D1-2-17_11(E120_S18_ID44)	120	18	0.164286	1787	48.2973
D5-1-17_1(E11_S3_ID98)	11	3	1.000000	*48	*1.1163
D5-1-17_2(E13_S3_ID99)	13	3	0.846154	*12	*0.2791
D5-1-17_3(E200_S34_ID5)	200	34	0.158945	3593	83.5581
D5-1-18_1(E9_S2_ID263)	9	2	1.000000	*8	*0.1633
D5-1-18_2(E13_S3_ID88)	13	3	0.846154	*12	*0.2449
D5-1-18_3(E14_S2_ID200)	14	2	0.736264	*10	*0.2041
D5-1-18_4(E223_S41_ID1)	223	41	0.118046	3215	65.6122
D5-2-17_1(E18_S1_ID199)	18	1	1.000000	*108	*2.5116
D5-2-17_2(E324_S42_ID1)	324	42	0.101307	8254	191.9535
D5-2-18_1(E18_S1_ID97)	18	1	1.000000	54	1.1489
D5-2-18_2(E56_S5_ID94)	56	5	0.318182	140	2.9787
D5-2-18_3(E345_S41_ID1)	345	41	0.116144	6425	136.7021
D5-3-18_1(E5_S2_ID40)	5	2	1.000000	*6	*0.1395
D5-3-18_2(E7_S1_ID59)	7	1	1.000000	*18	*0.4186
D5-3-18_3(E118_S40_ID3)	118	40	0.097349	1382	32.1395
D6-1-18_1(E12_S1_ID470)	12	1	1.000000	*7	*0.1228
D6-1-18_2(E22_S2_ID85)	22	2	0.636364	*32	*0.5614
D6-1-18_3(E403_S52_ID1)	403	52	0.092947	9754	171.1228
D6-2-18_1(E14_S1_ID1)	14	1	1.000000	*1	*0.0175
D6-2-18_2(E22_S1_ID343)	22	1	1.000000	*56	*0.9825
D6-2-18_3(E493_S54_ID3)	493	54	0.077904	7826	137.2982

$$\begin{aligned} \min \quad & 16 * \sum_{v_1,v_2 \in \mathbb{E}} W_{v_1,v_2} * y16_{v_1,v_2} + 8 * \sum_{v_1,v_2 \in \mathbb{E}} W_{v_1,v_2} * y8_{v_1,v_2} \\ & +4 * \sum_{v_1,v_2 \in \mathbb{E}} W_{v_1,v_2} * y4_{v_1,v_2} + 2 * \sum_{v_1,v_2 \in \mathbb{E}} W_{v_1,v_2} * y2_{v_1,v_2} \\ & \qquad + \sum_{v_1,v_2 \in \mathbb{E}} W_{v_1,v_2} * y1_{v_1,v_2} \end{aligned} \tag{3}$$

$$\text{s.t.} \quad v_1 \neq v_2 \quad \forall (v_1, v_2) \in \mathbb{E} \tag{4}$$

$$\begin{aligned} y8_{v_1,v_2} &= (v_1 - v_2 = 2) + (v_1 - v_2 = -2) \quad \forall (v_1, v_2) \in \mathbb{E} \\ y4_{v_1,v_2} &= (v_1 - v_2 = 3) + (v_1 - v_2 = -3) \quad \forall (v_1, v_2) \in \mathbb{E} \\ y2_{v_1,v_2} &= (v_1 - v_2 = 4) + (v_1 - v_2 = -4) \quad \forall (v_1, v_2) \in \mathbb{E} \\ y1_{v_1,v_2} &= (v_1 - v_2 = 5) + (v_1 - v_2 = -5) \quad \forall (v_1, v_2) \in \mathbb{E} \end{aligned} \tag{5}$$

$$y16_{v_1,v_2} + y8_{v_1,v_2} + y4_{v_1,v_2} + y2_{v_1,v_2} + y1_{v_1,v_2} \leq 1 \quad \forall (v_1, v_2) \in \mathbb{E} \tag{6}$$

Finally, let $\mathbb{I}^+$ be the set of sets of interchangeable examinations as defined in [8]. In order to break a symmetry of the problem we enforce an order over the examinations belonging to each set. This is formulated in Equation 7, where members of each set $\mathbb{S}$ of the sets in $\mathbb{I}^+$ are ordered among each other.

$$v_i \leq v_{i+1} \quad \forall v_i \in \mathbb{S} : i \in 1 \ldots |\mathbb{S}| - 1, \quad \forall \mathbb{S} \in \mathbb{I}^+ \tag{7}$$

Other formulations of the mathematical model have been proposed in the past. An example is the work in [4] that uses the so-called channeling constraints that were originally proposed in [1]. A difference in our model is that we employ the concept of interchangeable examinations that are embedded in the formulation. Moreover, the objective function is constructed equivalently, but differently, in our case.

5.2 Intelligent enumeration

Some of the instances have a comparatively small number of available periods. It's noteworthy that even small sub-problems with a few periods and a relatively low number of examinations are hard to optimally solve by current state of the art mixed integer programming solvers. A new method was developed to handle instances, and this method depending on the number of examinations, available periods and the conflict density of the corresponding graph is able to solve some problems to optimality. Moreover, the same method can be exploited and reach good solutions for bigger instances.

To best describe this process we will use a toy example with its graph representation pictured in Figure 1. Let the available periods for this problem to be four. The problem consists of five examinations with a varying number of common students between certain pairs of exams. Note that exams $1, 2, 3$ form a non trivial clique e.g. they are a complete sub-graph of the graph. As no student is allowed to participate in more than one examination per period, those three examinations will end up in three different periods. Also, examination 5 with a weighted degree of just 2 doesn't seem to play a major part in the grander scheme of things.

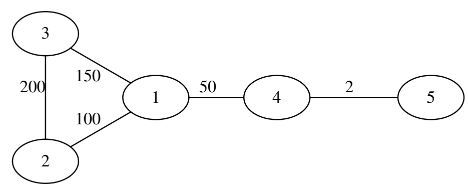

Fig. 1. Toy example for demonstrating the intelligent enumeration scheme.

The method can be used to either search for a good solution or to prove optimality, based on characteristics of the problem in question. The main idea remains the same for both cases. Firstly, we reduce the problem size by removing some of its exams. Then, we generate partial solutions, evaluate their cost and if it falls under some cut-off limit, which could be the cost of the best known solution, we fill the missing examinations to form a complete solution. This process is expected to act as a filter and has the potential to be computationally faster than a full enumeration.

The main idea of the method is to exploit a clique in the graph. In selecting a clique, it usually makes sense to choose the maximum clique. In the toy example, the maximum clique is the set of examinations $\{1, 2, 3\}$. It is guaranteed that the clique's examinations will end up on different periods which, for convenience, we name after them, $\{P_1, P_2, P_3\}$ correspondingly. Since we have four available periods we will name the period that will not be occupied by any of them as P_E. The remaining examinations $\{4, 5\}$ can be easily checked in this small example about their possible final positions. So, examination 4 can be placed in any of $\{P_2, P_3, P_E\}$ and examination 5 can join any period $\{P_1, P_2, P_3, P_E\}$.

Since examinations for the clique are fixed in periods $\{P_1, P_2, P_3\}$ the possible assignments for examinations 4 and 5 are $(4 : P_2, 5 : P_1), (4 : P_2, 5 : P_3), (4 : P_2, 5 : P_E), (4 : P_3, 5 : P_1), (4 : P_3, 5 : P_2), (4 : P_3, 5 : P_E), (4 : P_E, 5 : P_1), (4 : P_E, 5 : P_2), (4 : P_E, 5 : P_3)$ while $(4 : P_2, 5 : P_2), (4 : P_3, 5 : P_3), (4 : P_E, 5 : P_E)$ are infeasible as examinations 4 and 5 are in conflict. In total, there are 9 feasible schedules. If we had opted to leave examination 5 out, there would be just 3 feasible schedules $(4 : P_2), (4 : P_3), (4 : P_E)$.

Initially, we ignore examination 5 and we examine all possible permutations of $\{P_1, P_2, P_3, P_E\}$. We complement every permutation with each of the 3 possible partial schedules $(4 : P_2), (4 : P_3), (4 : P_E)$. Since each schedule and its reverse have exactly the same objective value, we can skip mirrored permutations, effectively cutting off half of the search space, thus eliminating this kind of symmetry. Nevertheless, for large numbers of periods, it is unrealistic to traverse all possible permutations, even by considering half of them. In the toy example, we evaluate $(4!/2) * 3$ partial solutions and we keep those that have cost under a cut-off barrier. The unscheduled examination 5 has a weighted degree of just 2, while other examinations have weighted degrees ranging from 52 to 350. So, most of the partial solutions should be filtered out.

Examination 5 of the toy example was initially ignored. A similar decision must be taken for each problem, about the examinations that will be initially ignored too. Unfortunately, this is not a trivial task. We cannot remove examinations of the chosen clique, should we wish to do so we should pick another clique. Intuitively, we want to initially ignore examinations with low degrees and weighted degrees, as they are able to appear in more periods. Consequently, they allow for more possible outcomes while at the same time their impact on the objective function is minor. It should be noted that not all partial solutions (solutions with ignored examinations still unscheduled) may lead to feasible solutions. So, for the case that full enumeration is unrealistic, quick feasibility checks can reveal unpromising partial solutions that are meaningless to be completed. The method is tuned by balancing the number of possible partial schedules generated with respect to the impact that the selected examinations have on the objective. The tuning is guided by selecting, through sampling, suitable examinations that will hopefully result in cutting-off many possible solutions. For the toy example the costs of these partial solutions are depicted in Table 3

Table 3. Permutations and partial solutions costs for the toy example in Fig. 1

P_1	P_2	P_3	P_E	$4 : P_2$	$4 : P_3$	$4 : P_E$
0	1	2	3	6800	6400	6200
0	1	3	2	4600	4000	4200
0	2	1	3	6800	7200	6600
0	2	3	1	5000	4800	5400
0	3	1	2	4600	5200	4800
0	3	2	1	5000	5200	5600
1	2	0	3	6400	6400	6000
1	2	3	0	6800	6400	6800
1	3	0	2	4400	4800	4800
1	3	2	0	6800	7200	7200
2	3	0	1	4400	4000	4400
2	3	1	0	6400	6400	6000

To further augment our filter while keeping computational cost low it's possible for partial solutions that are under the cut-off barrier to calculate the minimum cost each unscheduled examination can possibly introduce to the partial solution. If the sum of those minimum costs plus our partial solutions cost is under the cut-off barrier, the partial solution may lead to a desired complete solution. This process can be seen as a multi-layer filter like the one depicted in Fig. 2.

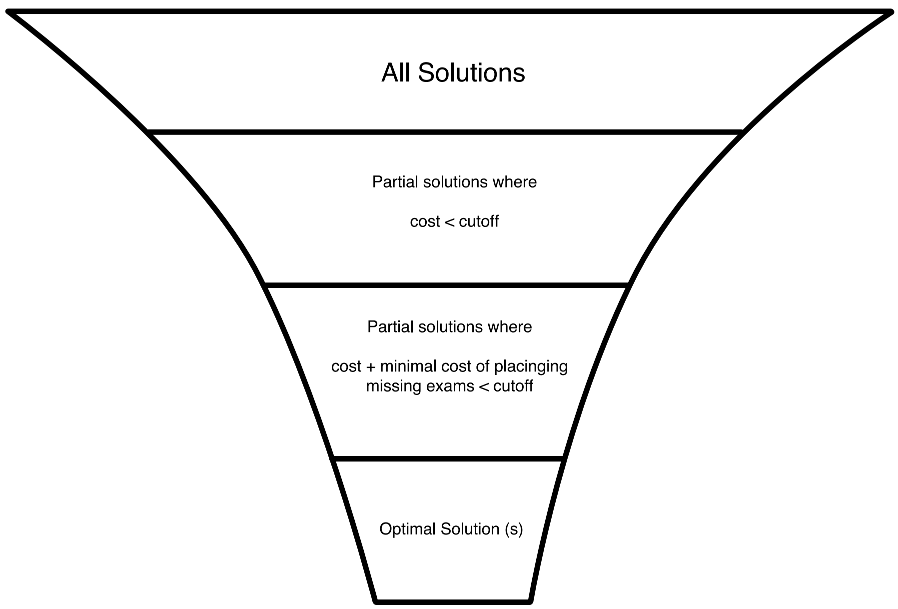

Fig. 2. Filter process.

5.3 Estimating lower bounds

Each students schedule is also an UETP sub-problem where his examinations are a complete graph where all edges have a weight of 1. This problem can be solved optimally for almost all instances, especially for those with a low number of periods.

Summing up those minimum penalties for all students can provide us with a lower bound. In the rare occasion that a solution's objective function is equal to this bound then this solution is optimal.

6 sta83 optimal solution

No optimality has ever been proved for any Carter's dataset instance until now. In this section we show that the solution for sta83 having value 95947 (95947/611=157.0327 in decimal value, where 611 is the total number of students for sta83) which appears in many papers is indeed optimal.

Instance sta83 consists of 139 exams, 13 periods and has a relatively low conflict density of value 0.14. The instance has no noise examinations and no noise students as defined in Section4. The instance is comprised of 3 disconnected components as shown in Fig. 3.

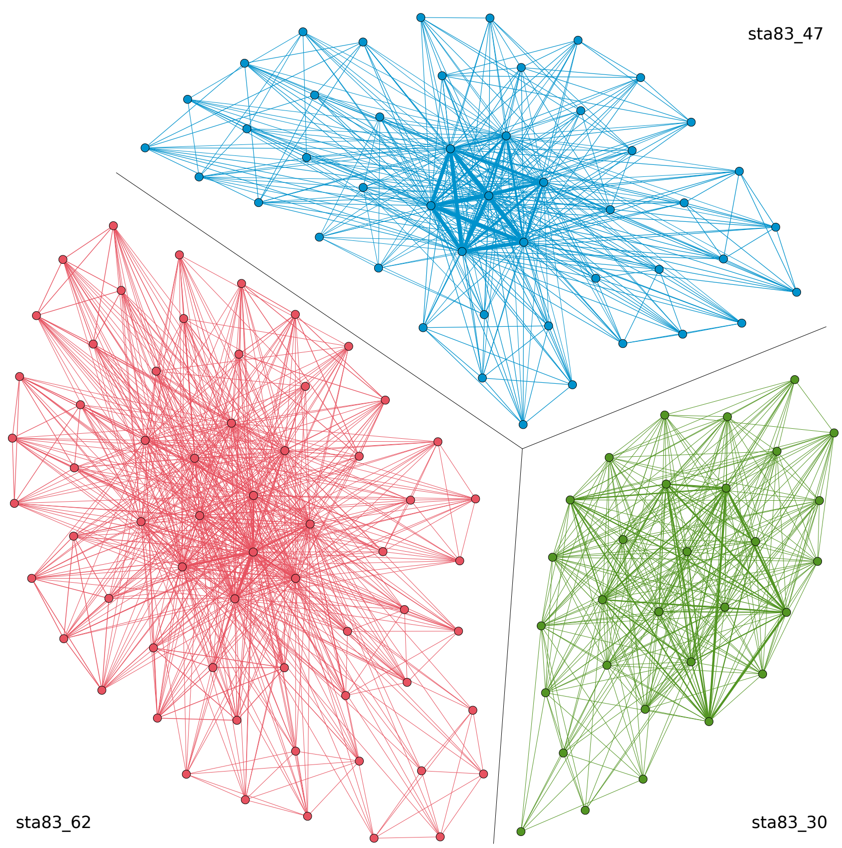

Fig. 3. Disconnected components of sta83. The weight of each edge is indicated by its thickness.

We can divide the problem into three independent subproblems because these components are disconnected. That is, there are three unique groups of students, each of which does not have an examination in common with the other two groups, allowing us to work on each component independently. The sum of these answers would be the optimal solution provided that all three of them are solved optimally. Motivated by the prospect of proving optimality for a Carter's dataset instance, we focused our attention on this task, and we managed to optimality solve each subproblem using a different approach, resulting in a novel way of handling high conflict-density components.

6.1 Component sta83_62

This is the largest component of sta83, having 62 examinations and a conflict density of 0.36. We tried to solve it using the model described in Section 5.1 using the IBM ILOG CPLEX IP solver. Unfortunately, after several hours the solver was unable to prove optimality. We tried to warm start the solution with the current best solution and have set the MIP emphasis parameter first to "emphasize optimality over feasibility" and then to "emphasize moving best bound". Both attempts were unsuccessful.

We noticed that the component has a special structure. It contains 10 sets of examinations with each set consisting of exactly 5 interchangeable examinations. These examinations amount for 50 of the 62 examinations that the component has in total. Details of these sets are presented in Table 4. Since interchangeable examinations can freely swap places with each other while keeping the objective value unchanged, the introduction of the symmetry breaking constraints of Equation 6 greatly improved the solver's efficiency in proving the optimal solution.

We also noticed that 3 examinations existed (72, 133, 136) in the graph that had connections with all other exams. So, we tried an approach that fixed these 3 examinations in specific periods and then tried to solve the remaining problem using IBM ILOG CPLEX. This time, the result was successful, the solver was able to return a result, either optimal or infeasible in a few minutes. It should be noted that infeasibility occurs because the cost of the best known solution is used as a cutoff constraint. So, we had only to try all possible places for positioning the 3 examinations and then solve the resulting problem. Since there are only 13 periods in instance sta83, this would mean that only $\binom{13}{3} = 286$ configurations existed that should be multiplied by $\frac{3!}{2}$ since the 3 examinations can occupy the fixed periods in any order (divided by 2 due to the inherent symmetry of the problem).

By exploiting the above observations, IBM ILOG CPLEX IP solver was able to solve each subproblem in a few minutes. After solving all subproblems, the optimal solution for sta83_62 was proved to be 32695. This solution occurred when examinations 72, 133 and 136 were fixed to periods 3, 6 and 8 respectively. The symmetric solution also exists and is produced by fixing examinations 72, 133 and 136 to periods 9, 6 and 4. Of course, many more symmetric solutions exist due to the interchangeable exams.

6.2 Component sta83_47

This component proved to be the easy part. It consists of 47 examinations and has a conflict density of 0.35. As described in subsection 5.3 we can estimate a lower bound by adding the minimum cost each student's schedule could possibly inflict. So, for each student in isolation, an IP model is formulated that given only the number of periods and the number of examinations that this student participates, decides about the schedule that results to the minimum possible cost. Of course, since each student is examined in isolation if two students share the same number of examinations then the problem needs to be solved just once. In practice, this is the case for several students. By adding minimum penalties of all students we have a lower bound for this component, which is 42750. The best known solution turns out to have cost equal to the lower bound obtained in this manner. Thus, the optimal solution for this component is 47250.

6.3 Component sta83_30

This was the last component to solve. It's the smallest one with just 30 examinations but a high conflict density of 0.72. With high hopes since just the smallest piece of the puzzle was missing, we were surprised to find out that to the best of our ability our MIP models were not able to prove an optimal solution. We have tried the same trick that we have used successfully in component sta83_62. We noticed that in the case of sta83_30 there is only one examination (134) that is connected to every other one. So, we tried to fix this examination to each period in turn and then to solve the remaining problems using IBM ILOG CPLEX. Unfortunately, this did not helped the solver to prove the optimality of the solution. Each subproblem seemed to run forever.

By observing closely the high density graph of this component we came up with the idea of separating examinations with high degrees and examinations with relatively low degrees. A similar idea has been exploited by [11] and others in constructing timetables giving precedence to high degree examinations and

Table 4. Component sta83_62, sets of interchangable examinations and their characteristics.

Set	Degree	Weighted Degree
{17, 38, 58, 85, 120}	8	8
{18, 39, 59, 86, 121}	16	240
{19, 40, 60, 87, 122}	16	264
{20, 41, 61, 88, 123}	15	168
{21, 42, 62, 89, 124}	12	88
{22, 43, 63, 90, 125}	16	160
{23, 44, 64, 91, 126}	15	160
{24, 45, 65, 92, 127}	16	264
{25, 46, 66, 93, 128}	16	280
{26, 47, 67, 94, 129}	16	280

leaving for a later phase the low degree ones. In our approach, we isolated the maximum clique, which for this particular instance comprises of 12 examinations and tried to arrange those examinations to the 13 periods leaving one period empty for each possible arrangement.

A significant observation is that irrelevant of the periods that the clique occupies, the possible placements for the remaining examinations will be the same because their possible positions are constrained by the examinations of the clique. By multiplying the number of those possibilities with the number of permutations of the periods we were able to count all possible solutions to be $13! * 109152$ where 13! is the number of possible period permutations and 109152 is the number of possible ways to schedule the remaining examinations for the specific component. This number is still quite large so we exploited the method described in Section 5.2. We aim to find a set of examinations that has minor impact on the cost but at the same time possible final positions of the sets' examinations might be disproportionate large. Fig. 4 which shows the degrees and weighted degrees of examinations was used as a visual aid for identifying the examinations needed. These examinations should reside at the lower left corner and should have the desirable characteristics. For sta83_30 a good set of examinations proved to be $\{5, 131, 28, 48, 76\}$ that manages to lower multiplier 109152 to just 47.

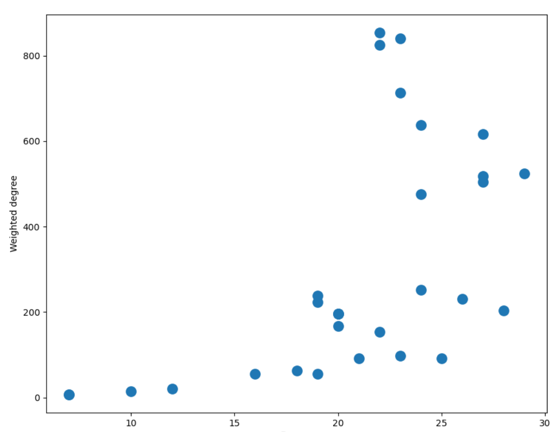

Fig. 4. Scatter plot of sta83_30 that gives insight about the set of examinations that should be scheduled last

The unscheduled examinations weighted degree is comparatively low and so the method has the potential of working effectively. By keeping in the set of initially unscheduled examinations, examinations that can easily move around the schedule, the number of possible partial solutions becomes quite low. Moreover, the low weighted degree that these examinations have prohibits from heavy impacts on the objective function. So, the filtering process is working. For the case of sta83_30 this "intelligent" search resulted in 13 distinct optimal solutions (and their symmetric ones) all having the same cost, 16002. The search was implemented in Julia [5] using its parallel computing features for the CPU. Five high end workstations were simultaneously running the experiment and the time needed was about 12 hours.

7 Conclusions

This work was about the uncapacitated examination timetabling problem. It continues previous work of our team. A key observation is that even for this rather simple scheduling problem that is only an abstraction of the corresponding real-life problem, the proof that a given solution is optimal is definitely not trivial. Nevertheless, our team succeeded in proving the optimality of a certain instance, namely sta83 of the Carter's dataset. In order for this to happen we had to decompose the problem into independent subproblems. Having 3 problems of moderate size gave us the opportunity of experimenting with various approaches. No method was able to solve all three subproblems. After many experiments and carefully analyzing the components, we finally discovered three approaches that were able to prove optimality. Each subproblem was solved by a different approach and the optimal solution for sta83 was proved. Furthermore, we contributed two new best solutions to public dataset problems, alongside with several optimal solutions to subproblems that exist in various instances.

Acknowledgements We acknowledge support of this work by the project "Dioni: Computing Infrastructure for Big-Data Processing and Analysis." (MIS No. 5047222) which is implemented under the Action "Reinforcement of the Research and Innovation Infrastructure", funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

References

1. Aardal, K.I., Van Hoesel, S.P., Koster, A.M., Mannino, C., Sassano, A.: Models and solution techniques for frequency assignment problems. Annals of Operations Research **153**(1), 79–129 (2007)
2. Alefragis, P., Gogos, C., Valouxis, C., Housos, E.: A multiple metaheuristic variable neighborhood search framework for the uncapacitated examination timetabling problem. In: Proceedings of the 13th International Conference on the Practice and Theory of Automated Timetabling-PATAT. vol. 1, pp. 159–171 (2021)

3. Battistutta, M., Ceschia, S., Cesco, F.D., Gaspero, L.D., Schaerf, A., Topan, E.: Local search and constraint programming for a real-world examination timetabling problem. In: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research. pp. 69–81. Springer (2020)
4. Bellio, R., Ceschia, S., Di Gaspero, L., Schaerf, A.: Two-stage multi-neighborhood simulated annealing for uncapacitated examination timetabling. Computers & Operations Research **132**, 105300 (Aug 2021)
5. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A fresh approach to numerical computing. SIAM Review **59**(1), 65–98 (2017). https://doi.org/10.1137/141000671
6. Carter, M.W., Laporte, G., Lee, S.Y.: Examination Timetabling: Algorithmic Strategies and Applications. Journal of the Operational Research Society **47**(3), 373–383 (Mar 1996)
7. Ceschia, S., Di Gaspero, L., Schaerf, A.: Educational timetabling: Problems, benchmarks, and state-of-the-art results. arXiv preprint arXiv:2201.07525 (2022)
8. Gogos, C., Dimitsas, A., Nastos, V., Valouxis, C.: Some insights about the uncapacitated examination timetabling problem. In: 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM). pp. 1–7. IEEE (2021)
9. Kristiansen, S., Stidsen, T.R.: A comprehensive study of educational timetabling, a survey. Department of Management Engineering, Technical University of Denmark.(DTU Management Engineering Report **8** (2013)
10. Qu, R., Burke, E.K., McCollum, B., Merlot, L.T., Lee, S.Y.: A survey of search methodologies and automated system development for examination timetabling. Journal of scheduling **12**(1), 55–89 (2009)
11. Rahman, S.A., Bargiela, A., Burke, E.K., Ozcan, E., McCollum, B.: Construction of examination timetables based on ordering heuristics. In: 2009 24th International Symposium on Computer and Information Sciences. pp. 680–685. Ieee (2009)
12. Schaerf, A.: A survey of automated timetabling. Artificial intelligence review **13**(2), 87–127 (1999)
13. Tan, J.S., Goh, S.L., Kendall, G., Sabar, N.R.: A survey of the state-of-the-art of optimisation methodologies in school timetabling problems. Expert Systems with Applications **165**, 113943 (2021)