Improved bounds and solutions for the invigilator assignment problem

Reinout Annaert[1]

KU Leuven, Department of Computer Science, NUMA, Belgium
reinout.annaert@kuleuven.be

Abstract. This paper addresses the exam invigilator assignment problem with interaction-oriented objectives in a university with multiple departments. The problem consists of assigning invigilators to exams, where each exam requires a given number of invigilators, with each invigilator subject to availability and workload constraints. We consider two objective variants. Contact maximization seeks to maximize the number of unique interdepartmental contacts to stimulate interaction between departments. Contact reduction seeks to minimize the total number of unique contacts to limit health risks during pandemics or seasonal outbreaks. For both variants, we develop bounds on the objective value and propose a multi-neighborhood local search heuristic. Computational experiments on benchmark instances show that our proposed bounds improve or match almost all previously reported bounds. Moreover, the proposed heuristic improves upon all previously reported best-known solutions. For many contact-maximization instances, the best solution found matches the new bound, thereby proving optimality.

Keywords: Exam invigilation · Timetabling · Interaction-oriented objectives.

1 Introduction

The exam invigilator assignment problem (IAP) concerns the assignment of invigilators to a set of exams subject to operational requirements such as exam coverage, invigilator availability and workload restrictions. Despite its practical relevance in university examination planning, literature on the IAP remains relatively limited. Only recently, Verplancke et al. [15] introduced publicly available benchmark instances, thereby enabling far more systematic computational studies to take place.

Beyond operational feasibility, invigilator assignments can also be used to influence interactions between invigilators. Recent work has therefore considered interaction-oriented objectives, which aim to influence which invigilators supervise exams together. Assigning invigilators from different departments to the same exam stimulates interdepartmental interaction, whereas limiting the number of distinct contacts is desirable when contact reduction is important, for example during the COVID-19 pandemic or seasonal outbreaks such as flu.

Verplancke et al. [15] formalize these two settings as contact maximization and contact reduction, and prove that both variants are NP-hard via reductions from the Social Golfer Problem [14] and Exact Cover by 3-Sets [5], respectively.

However, further opportunities remain to strengthen our computational understanding of these interaction-oriented variants. This paper addresses two important challenges: strengthening the best available bounds for both variants to enable a more accurate assessment of heuristic solution quality, and developing an improved local search heuristic.

The remainder of the paper is organized as follows. Section 2 reviews the related literature. Next, Section 3 formally defines the IAP with interaction-oriented objectives. Section 4 presents our new bounds for both variants, while Section 5 introduces our proposed local search heuristic. Section 6 reports the computational study. Finally, Section 7 concludes the paper.

2 Related work

The academic literature on the IAP is dominated by institution-specific case studies rather than benchmark-driven research. Across these studies, the focus is on constructing complete invigilation schedules for given examination timetables that respect specific institutional rules.

A common theme in the literature is the modeling of *fairness* and *preferences*. Fairness is often incorporated in the objective by minimizing dissatisfaction or balancing workload [1,4,8,9,10,12,13], or it can also be enforced through workload bounds or balancing constraints [3,11,15]. Preferences are treated similarly: some studies include them directly in the objective – through penalties for undesirable time slots, or preference-based assignment scores [1,8,9,12,13,15] – while other studies model them as constraints [3,4,11].

Beyond these modeling choices, previous research also differs in terms of the solution methods used. Several papers formulate the IAP as a mixed-integer programming (MIP) model [1,3,12,13,15]. These models are often combined with heuristic decomposition or variable-fixing techniques to obtain solutions within reasonable computation times. Other studies rely on constructive heuristics [9,10,11], goal programming [4] or heuristics such as scatter search [8] to generate solutions.

Despite this methodological variety, little attention has been paid to computing bounds, making it difficult to accurately assess solution quality. Moreover, the lack of public benchmark data has limited comparability, as most studies rely on proprietary instances from a single institution. As a result, meaningful comparisons between methods have remained scarce. Verplancke et al. [15] address this limitation by introducing public benchmark instances for the IAP, thereby enabling systematic comparison.

3 Problem description

The IAP with interaction-oriented objectives involves a set of exams E, invigilators I, timeslots T and departments K. Each exam $e \in E$ is scheduled during a timeslot $t_e \in T$ and requires d_e invigilators, where $d_e \in \mathbb{Z}_{\geq 0}$.

Each invigilator $i \in I$ is available during a subset of timeslots $A_i \subseteq T$ and belongs to a department $k_i \in K$. Furthermore, each invigilator must be assigned to a number of exams between a minimum and a maximum workload:

$$\underline{w}_i,\ \overline{w}_i \in \mathbb{Z}_{\geq 0}, \qquad \underline{w}_i \leq \overline{w}_i, \tag{1}$$

where $\underline{w}_i$ and $\overline{w}_i$ denote the minimum and maximum number of exams to which invigilator i must be assigned, respectively.

A feasible solution assigns exactly d_e available invigilators to each exam $e \in E$. An invigilator cannot be assigned to more than one exam in the same timeslot, and all workload bounds must be satisfied.

Interaction is modeled at two levels: *co-assignments* and *contacts*. Two distinct invigilators i and j are *co-assigned* to exam e if both are assigned to e. They have a *contact* if they are co-assigned to at least one exam. Thus, a contact is a unique unordered pair $\{i, j\}$, counted at most once regardless of how many exams invigilators i and j are co-assigned to. A contact $\{i, j\}$ is considered a *interdepartmental contact* if $k_i \neq k_j$.

We study two variants of the IAP. In contact maximization (IAP-CM), the goal is to maximize the number of interdepartmental contacts. In contact reduction (IAP-CR), the goal is to minimize the total number of contacts, regardless of whether they are interdepartmental or intradepartmental.

4 Improved bounds

To derive new bounds, we will use the following binary variables for assignments and contacts. Let $x_{i,e}$ be 1 if invigilator $i \in I$ is assigned to exam $e \in E$, and 0 otherwise. For each pair of invigilators $i, j \in I$ with $i < j$ and exam $e \in E$, let $w_{ij,e}$ be 1 if i and j are co-assigned to exam e, and 0 otherwise, and let y_{ij} be 1 if they have a contact, and 0 otherwise.

Using this notation, the objective of IAP-CM is

$$\max \sum_{\substack{i,j \in I \\ i<j,\ k_i \neq k_j}} y_{ij}, \tag{2}$$

whereas the objective of IAP-CR is

$$\min \sum_{\substack{i,j \in I \\ i<j}} y_{ij}. \tag{3}$$

4.1 Upper bounds for IAP-CM

The demand-only upper bound, introduced by [15], is based solely on the number of invigilators needed for an exam. For an exam e with demand d_e, at most $\binom{d_e}{2}$ unordered invigilator pairs can be co-assigned. Since each interdepartmental contact requires at least one co-assignment to an exam, the total number of interdepartmental contacts cannot exceed the total number of co-assigned pairs across all exams:

$$\sum_{e \in E} \binom{d_e}{2}. \tag{4}$$

To obtain a stronger bound, we enforce an upper bound on objective (2) by counting all interdepartmental co-assignments instead of all interdepartmental contacts. Since each contact corresponds to at least one co-assignment, this yields a valid upper bound. For each exam $e \in E$ and department $k \in K$, let us define

$$z_{k,e} = \sum_{i:\, k_i = k} x_{i,e}. \tag{5}$$

Thus, $z_{k,e}$ denotes the number of invigilators from department k assigned to exam e. For exam e, the total number of co-assigned pairs equals $\binom{d_e}{2}$, while the number of intradepartmental co-assigned pairs equals $\sum_{k \in K} \binom{z_{k,e}}{2}$. This yields the objective used to compute the upper bound:

$$\max \sum_{e \in E} \left(\binom{d_e}{2} - \sum_{k \in K} \binom{z_{k,e}}{2} \right). \tag{6}$$

Maximizing (6) is equivalent to minimizing the number of intradepartmental co-assigned pairs. Let us define:

$$C^\star = \min \sum_{e \in E} \sum_{k \in K} \binom{z_{k,e}}{2}. \tag{7}$$

Then the optimal value of the objective equals:

$$\mathrm{UB}_2 = \sum_{e \in E} \binom{d_e}{2} - C^\star. \tag{8}$$

Minimization problem (7) can be solved using a min-cost flow network. The network sends flow from invigilators to exams through timeslots and departments. A feasible flow corresponds to a solution satisfying workload, availability and exam demand constraints. Costs are incurred only on department–exam arcs and represent intradepartmental co-assigned pairs.

To model the network, we introduce the following auxiliary quantities. For each timeslot $t \in T$, let

$$E_t = \{e \in E : t_e = t\}, \tag{9}$$

and set $\delta_t = 1$ if $E_t \neq \emptyset$, and 0 otherwise.

We define the effective maximum workload of invigilator i as follows:

$$\overline{w}_i^{\mathrm{eff}} = \min\left\{\overline{w}_i,\ \sum_{t\in A_i}\delta_t\right\} \qquad \forall i \in I. \tag{10}$$

For each department $k \in K$ and timeslot $t \in T$, let

$$n_{k,t} = \left|\{\, i \in I : k_i = k,\ t \in A_i \,\}\right| \tag{11}$$

denote the number of invigilators from department k available during timeslot t. For each exam $e \in E$ and department $k \in K$, let us define

$$m_{k,e} = \min\{d_e,\ n_{k,t_e}\} \tag{12}$$

as the maximum number of invigilators from department k that can be assigned to exam e.

The network's node types are summarized in Table 1. Let $D = \sum_{e\in E} d_e$ denote total exam demand. Node s represents the source supplying the D units of flow corresponding to invigilator assignments, while node f represents the sink collecting those D units of flow required to satisfy exam demand. For each invigilator $i \in I$, node v_i aggregates the assignments of invigilator i. For each available invigilator–timeslot pair (i,t), node $u_{i,t}$ ensures that invigilator i can be assigned to at most one exam during timeslot t. Nodes $p_{k,e}$ aggregate assignments of invigilators from department k to exam e, while node q_e collects all invigilators assigned to exam e. The arc types are summarized in Table 2, where ℓ and u denote the lower and upper bounds for each arc.

Table 1. Node types in the min-cost flow network.

Node	Exists for
s, f	–
v_i	$\forall i \in I$
$u_{i,t}$	$\forall i \in I,\ \forall t \in A_i,\ E_t \neq \emptyset$
$p_{k,e}$	$\forall e \in E,\ \forall k \in K:\ n_{k,t_e} > 0$
q_e	$\forall e \in E$

If $z_{k,e}$ invigilators from department k are assigned to exam e, they form $\binom{z_{k,e}}{2}$ intradepartmental pairs. This is modeled using $m_{k,e}$ parallel arcs from $p_{k,e}$ to q_e with incremental costs $0, 1, \ldots, m_{k,e}-1$, so that sending $z_{k,e}$ units of flow incurs the following cost:

$$\sum_{j=1}^{z_{k,e}}(j-1) \;=\; \binom{z_{k,e}}{2}. \tag{13}$$

Solution method. We solve the min-cost flow problem with lower and upper bounds by applying the standard transformation that eliminates lower bounds

Table 2. Arc types in the min-cost flow network.

Arc	Exists for	ℓ	u	Cost
$(s \to v_i)$	$\forall i \in I$	$\underline{w}_i$	$\overline{w}_i^{\text{eff}}$	0
$(v_i \to u_{i,t})$	$\forall i \in I,\ \forall t \in A_i,\ E_t \neq \emptyset$	0	1	0
$(u_{i,t} \to p_{k_i,e})$	$\forall (i,t),\ \forall e \in E_t$	0	1	0
$(p_{k,e} \to q_e)$	$\forall e \in E,\ \forall k \in K,\ j = 1, \dots, m_{k,e}$	0	1	$j-1$
$(q_e \to f)$	$\forall e \in E$	d_e	d_e	0

and introduces node balances. Let (p,q) denote an arc in the network and let b_v denote the balance of node v, that is, the net outflow at node v. Initially, the balances are $b_s = D, \quad b_f = -D$, and $b_v = 0$ for all other nodes. The transformation then proceeds as follows:

1. Replace each arc capacity with $u_{pq} - \ell_{pq}$.
2. Adjust node balances $b_p \leftarrow b_p - \ell_{pq}$ and $b_q \leftarrow b_q + \ell_{pq}$.
3. Add arc (f,s) with sufficiently large capacity and zero cost.
4. Introduce an auxiliary super-source and super-sink. Connect nodes with $b_v < 0$ to the super-source and nodes with $b_v > 0$ to the super-sink.

We then solve the resulting minimum-cost flow problem using successive shortest augmenting paths with reduced costs. All capacities and balances are integral, hence an optimal integral flow exists. The optimal flow value equals Equation (7), thereby yielding upper bound UB_2.

4.2 Lower bounds for IAP-CR

Let us begin by recalling the lower bound proposed by [15] for IAP-CR. Their bound is based on two observations. First, supervising an exam with demand d_e creates exactly $\binom{d_e}{2}$ co-assigned invigilator pairs, while assigning invigilators in a new timeslot may force additional contacts with invigilators encountered earlier. Second, any feasible solution must use a sufficiently large number of distinct invigilators to cover total demand while satisfying workload constraints. The bound is constructed by processing timeslots iteratively. Given the timeslots already processed, each step selects the next timeslot so as to maximize the increase in the bound, given the timeslots already processed. This yields a valid lower bound, which we denote by LB_1:

$$\sum_{\substack{i,j \in I \\ i<j}} y_{ij} \ \geq \ \text{LB}_1. \tag{14}$$

To derive an alternative lower bound, we compare the total number of co-assigned pairs that must be created when supervising the exams with the maximum number of co-assignments that can be realized through a single contact. This yields a covering bound over contacts.

Supervising exam e with demand d_e creates exactly $\binom{d_e}{2}$ unordered co-assigned invigilator pairs. Hence any feasible solution induces a total of Equation (4) co-assigned pairs across all exams.

We can bound the contribution of a single contact. Indeed, an invigilator can be assigned to at most one exam per timeslot, while a pair $\{i, j\}$ can be co-assigned to at most one exam per timeslot during which both invigilators are available. Workload bounds also limit how often each invigilator can be assigned. We define the capacity of contact $\{i, j\}$ as follows:

$$u_{ij} = \min\left\{ \sum_{t \in A_i \cap A_j} \delta_t,\, \overline{w}_i^{\text{eff}},\, \overline{w}_j^{\text{eff}} \right\} \qquad \forall\, i, j \in I:\ i < j. \tag{15}$$

This quantity is an upper bound on the number of co-assignments that contact $\{i, j\}$ can realize in any feasible solution. If $\mathcal{M} = \{\{i, j\} : y_{ij} = 1\}$ denotes the set of contacts in a feasible solution, then the total number of co-assignments generated by these contacts is at most

$$\sum_{\{i,j\} \in \mathcal{M}} u_{ij}. \tag{16}$$

Feasibility therefore requires that these capacities cover all co-assigned pairs induced by the exams, namely:

$$\sum_{\{i,j\} \in \mathcal{M}} u_{ij} \ \geq\ \sum_{e \in E} \binom{d_e}{2}. \tag{17}$$

Let $\text{CoverLB}(r)$ denote the minimum number of contacts whose capacities sum to at least r. Since each contact contributes at most u_{ij} co-assignments, every feasible solution of the IAP-CR must satisfy

$$\sum_{\substack{i,j \in I \\ i < j}} y_{ij} \ \geq\ \text{CoverLB}\left(\sum_{e \in E} \binom{d_e}{2} \right). \tag{18}$$

We therefore define the lower bound as follows:

$$\text{LB}_2 = \text{CoverLB}\left(\sum_{e \in E} \binom{d_e}{2} \right). \tag{19}$$

Solution method. Computing LB_2 reduces to selecting the smallest number of contacts whose capacities u_{ij} cover $\sum_{e \in E} \binom{d_e}{2}$. Since all contacts have unit cost, this problem is solved optimally by sorting values u_{ij} in nonincreasing order and selecting them until the required capacity is reached.

5 Local search heuristic

To quickly compute solutions for both the IAP-CM and IAP-CR, we will develop a multi-neighborhood local search heuristic. Its purpose is to converge quickly to high-quality solutions and ultimately improve upon the best-known results for the benchmark instances.

5.1 Initial solution

We start the search from an empty solution in which no invigilators are assigned to exam slots. In this initial state, exam demands are unmet and invigilator workloads may fall below their required lower bounds. To allow the search to start from this infeasible state, violations of constraints are penalized in the objective function with large penalty terms. As a result, the local search gradually repairs the infeasible solution during the search.

5.2 Neighborhoods

The local search procedure uses a diverse set of neighborhoods. At each iteration, one neighborhood is selected at random and a candidate move is generated. Candidate moves are evaluated using the Late-Acceptance Hill Climbing (LAHC) acceptance criterion [2], with history length H. Some neighborhoods perform small local changes, while others reconstruct larger parts of the solution.

We can group each of the neighborhoods into one of three categories: reassignment, swap and reconstruction. Reassignment neighborhoods modify a single assignment, swap neighborhoods exchange assignments between exams, and reconstruction neighborhoods rebuild larger parts of the solution.

Reassignment neighborhoods

Simple reassign. The invigilator assigned to an exam is replaced by another invigilator who is available for the corresponding timeslot.

Contact-guided reassign. The invigilator assigned to an exam is replaced by another invigilator based on the objective. The available candidate that maximizes interdepartmental contacts (or minimizes contacts) is selected.

Swap neighborhoods

Timeslot swap. Two invigilators assigned to different exams in the same timeslot are swapped.

Cross-timeslot swap. Two invigilators assigned to different exams in different timeslots are swapped.

Three-way swap. Three invigilators assigned to three different exams in the same timeslot are cyclically swapped.

Contact-guided swap. All possible swaps between pairs of invigilators assigned to different exams in the same timeslot are considered as candidate moves. Each candidate swap is evaluated based on the contacts consequently created with the other invigilators.

Reconstruction neighborhoods

Ruin-and-recreate. All invigilators assigned to an exam are removed, after which the exam is greedily reconstructed.

Within-timeslot JV. One invigilator is removed from each exam in a randomly selected set of exams taking place during the same timeslot. The removed invigilators are then reassigned to the vacant positions by solving a linear assignment problem using the Jonker–Volgenant (JV) algorithm [6]. The number of reassigned invigilators, and thus the size of the assignment matrix, is limited by parameter $S_{\max}$.

Cross-timeslot JV. One invigilator is removed from each exam in a randomly selected set of exams. The removed invigilators are then reassigned to the vacant positions by solving a linear assignment problem using the JV algorithm, again limited to at most $S_{\max}$ reassigned invigilators.

6 Computational experiments

We evaluate our approach on the benchmark instances introduced by [15]. The experiments were performed on a machine with an AMD Ryzen 7 7800X3D processor and 32GB of RAM. We implemented the local search heuristic in Rust and compiled it in release mode.

We tuned two algorithm parameters automatically using the `irace` package [7]: history length H of the Late-Acceptance Hill Climbing acceptance criterion and the maximum size $S_{\max}$ of the JV assignment matrix. The tuning procedure consisted of 300 runs on all 17 instances and selected $H = 20$ and $S_{\max} = 15$, which we used in all experiments.

We compare our bounds and solutions against those of Verplancke et al. [15], who propose a multi-stage heuristic that combines a MIP model with two improvement phases. They first construct an initial solution by solving a polynomial-time MIP model for a preference-based variant of the IAP, using modified objective coefficients to indirectly incentivize either contact maximization or contact reduction. They then improve this solution through a swapping phase. Finally, they apply a fix-and-optimize heuristic in which they iteratively relax subsets of variables (such as those related to exams, timeslots, or invigilators) while fixing the remaining variables, before reoptimizing the resulting subproblems.

Since both approaches are stochastic, we performed five independent runs per instance for each approach, with a time limit of 600 seconds. For the approach of [15], we used the publicly available source code provided by the authors. We report both the average and the best objective value obtained across these runs.

The results for the IAP-CM in Table 3 show that our new upper bound UB_2 is significantly stronger than the previously reported bound UB_1. More precisely,

UB_2 improves UB_1 for 15 of the 17 instances and matches it for the remaining two (`ITC3` and `UG4`). Our heuristic matches or improves all previously reported best-known solutions. Indeed, for 15 of the 17 instances, the best solution found matches UB_2, thereby proving optimality. The results are also highly stable: all five runs attain the best reported value for 16 of the 17 instances.

Table 3. Comparison of bounds and solution methods for the IAP-CM variant.

Inst.	UB_1	UB_2	Obj_1		Obj_2		
			Avg	Best	Avg	Best	
ITC1	2,992	2,457	2,434.0	2,438	2,457.0	2,457	⊛
ITC2	4,172	4,137	4,106.2	4,107	4,137.0	4,137	⊛
ITC3	4,558	4,558	4,524.8	4,528	4,558.0	4,558	⊛
ITC4	1,377	1,325	1,310.6	1,314	1,325.0	1,325	⊛
ITC5	5,110	4,871	4,848.0	4,848	4,871.0	4,871	⊛
ITC6	2,030	1,912	1,892.6	1,895	1,912.0	1,912	⊛
ITC7	9,257	8,175	8,067.2	8,070	8,172.4	8,175	⊛
ITC8	5,015	4,262	4,234.6	4,240	4,262.0	4,262	⊛
ITC9	1,386	1,133	1,129.8	1,131	1,133.0	1,133	⊛
ITC10	1,791	1,524	1,523.4	1,524	1,524.0	1,524	⊛
ITC11	12,026	11,079	11,012.8	11,013	11,079.0	11,079	⊛
ITC12	988	779	779.0	779	779.0	779	⊛
UG1	79	78	78.0	78	78.0	78	⊛
UG2	1,603	1,507	1,477.0	1,477	1,507.0	1,507	⊛
UG3	1,178	1,098	1,079.0	1,087	1,097.0	1,097	
UG4	704	704	703.4	704	704.0	704	⊛
UG5	1,593	1,495	1,468.4	1,474	1,493.0	1,493	

UB_1, Obj_1: [15]; UB_2, Obj_2: this paper; ⊛ proven optimum.

Table 4. Comparison of bounds and solution methods for the IAP-CR variant.

Inst.	LB_1	LB_2	Obj_1		Obj_2	
			Avg	Best	Avg	Best
ITC1	224	365	1,669.0	1,651	1,285.2	1,274
ITC2	478	786	2,647.2	2,637	2,206.8	2,187
ITC3	773	1,140	3,228.0	3,225	2,749.2	2,745
ITC4	336	459	989.6	979	726.4	718
ITC5	648	900	2,898.4	2,896	2,659.6	2,629
ITC6	697	677	1,340.4	1,336	1,111.0	1,100
ITC7	727	1,277	5,440.2	5,429	5,132.8	5,076
ITC8	445	773	3,091.0	3,080	2,552.0	2,507
ITC9	329	347	910.6	903	747.6	726
ITC10	328	359	964.6	950	855.6	835
ITC11	1,492	1,718	6,731.4	6,697	6,443.6	6,402
ITC12	658	494	789.8	782	752.0	732
UG1	17	27	39.8	38	39.6	38
UG2	219	401	875.4	866	717.4	679
UG3	147	295	557.6	522	553.6	517
UG4	134	176	380.4	371	314.2	311
UG5	228	399	862	822	727.6	708

UB_1, Obj_1: [15]; UB_2, Obj_2: this paper; ⊛ proven optimum.

For the IAP-CR, Table 4 shows that our new lower bound LB_2 improves upon LB_1 for 15 of the 17 instances, often by a significant margin. The exceptions are `ITC6` and `ITC12`, where LB_1 remains stronger. Our heuristic improves the best-known solution for 16 of the 17 instances. Compared with IAP-CM, the results for the IAP-CR exhibit larger gaps between bounds and best solutions, as well as larger differences between average and best run values. This indicates that the minimization variant is more computationally challenging for both bounding and local search. We hypothesize that this is due to the fact that department information cannot be exploited in this variant.

Figures 1 and 2 show the aggregated convergence behavior of the proposed heuristic compared to [15]. For each instance, both methods were run five times with time limits of 10 seconds, 1 minute, 2 minutes, 10 minutes, 20 minutes and 1 hour. The reported gap is computed with respect to the best-known solution for each instance. The solid line represents the average gap over time, while the shaded band indicates one standard deviation. For the IAP-CM, our method

Fig. 1. Average gap over time (IAP-CM).

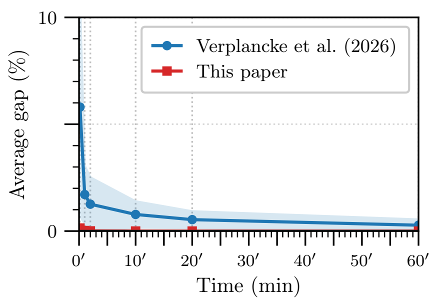
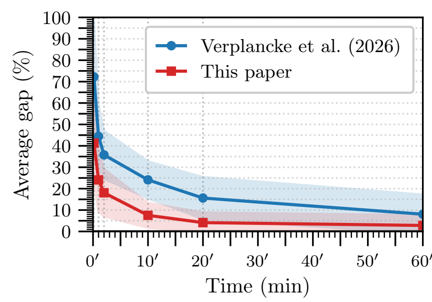


Fig. 2. Average gap over time (IAP-CR).



exhibits a very rapid improvement of the objective within the first few seconds. In fact, all instances except `ITC3` and `ITC11` reach a gap of 0% within 10 seconds. After one minute, all instances have a gap of 0% across all five runs. In contrast, the approach of [15] reduces the gap much more gradually and requires substantially more time to reach solutions of comparable quality.

For the IAP-CR, a different trend can be observed. Convergence is more gradual and takes longer, once again confirming that the minimization variant is more challenging for the heuristic. Nevertheless, our approach consistently outperforms [15], achieving both faster improvements in the early stages and better final solution qualities.

7 Conclusions and future directions

This paper strengthens the computational understanding of the invigilator assignment problem with interaction-oriented objectives. For both interdepartmental contact maximization and contact reduction variants, we (i) improve most existing bounds and (ii) improve the previously reported best-known solutions on the benchmark instances.

Our computational results also highlight a clear difference between the two variants. For the IAP-CM, our new upper bound is strong and the heuristic performs very well, enabling optimality to be proved for 15 out of 17 instances. For the IAP-CR, the proposed lower bound and heuristic solutions lead to significant improvements in both bounds and solution quality, but the gaps that remain indicate that this variant is more computationally challenging.

Several directions for future research remain. First, it would be worth investigating which instance characteristics drive computational difficulty in both variants. In particular, for the IAP-CM it would be interesting to study how the distribution of invigilators over departments affects both the quality of the bounds and the difficulty of finding good solutions. A second direction for future research is to investigate more flexible problem variants in which workload bounds and exam demands are treated as soft constraints rather than strict feasibility requirements.

Acknowledgments The author would like to thank Greet Vanden Berghe and Pieter Smet for proofreading, and Connolly Editorial for editorial consultation.

References

1. Al-Yakoob, S.M., Sherali, H.D., Al-Jazzaf, M.: A mixed-integer mathematical modeling approach to exam timetabling. Computational Management Science 7(1), 19–46 (2010)
2. Burke, E.K., Bykov, Y.: The late acceptance hill-climbing heuristic. European Journal of Operational Research 258(1), 70–78 (2017)
3. Güler, M.G., Geçici, E., Köroğlu, T., Becit, E.: A web decision support system for examination timetabling. Expert Systems with Applications 183, 115363 (2021)
4. Hanum, F., Romliyah, M.A., Bakhtiar, T.: Exam invigilators assignment problem: a goal programming approach. Applied Mathematical Sciences 9(58), 2871–2880 (2015)
5. Garey, M.R., Johnson, D.S.: Computers and intractability: A guide to the theory of NP-completeness. W.H. Freeman (1979)
6. Jonker, R., Volgenant, A.: Improving the Hungarian assignment algorithm. Operations Research Letters 5(4), 171–175 (1986)
7. López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., Stützle, T.: The irace package: Iterated racing for automatic algorithm configuration. Operations Research Perspectives 3, 43–58 (2016)
8. Lourenço, H.R., Martí, R., Laguna, M.: Assigning proctors to exams with scatter search. Universitat Pompeu Fabra (2001)
9. Matcı, D.K., Acar, I.: Modelling and optimization of the exam invigilator assignment problem based on preferences. Turkish Journal of Engineering 3(3), 140–148 (2019)
10. Mohmad Kahar, M.N., Kendall, G.: Universiti Malaysia Pahang examination timetabling problem: scheduling invigilators. Journal of the Operational Research Society 65(2), 214–226 (2014)
11. Othman, J., Wahab, A., Kadar, R., NizamWarris, S.: Assigning Examination Invigilator's Schedule Problem: An Approach of Programming Techniques Using Simple Sequential Assignment. JCRINN 2, 1–13 (2017)
12. Ozturk, Z.K., Ozturk, G., Sagir, M.: An automated multi-objective invigilator-exam assignment system. International Journal of Information Technology & Decision Making 9(2), 223–238 (2010)
13. Sagir, M., Ozturk, Z.K.: Exam scheduling: Mathematical modeling and parameter estimation with the Analytic Network Process approach. Mathematical and Computer Modelling 52(5–6), 930–941 (2010)
14. Triska, M.: Solution methods for the social golfer problem. Master's thesis, TU Wien, Austria (2008)
15. Verplancke, H., Van Bulck, D., Limère, V., Schmid, N.A.: The (a) social exam invigilator assignment problem. Journal of Scheduling (2026)