

Journal of the Operational Research Society

ISSN: 0160-5682 (Print) 1476-9360 (Online) Journal homepage: www.tandfonline.com/journals/tjor20

Improving the ranking position from the linear ordering problem with a limited budget

Iñigo Martín Melero, Javier Alcaraz & Mercedes Landete

To cite this article: Iñigo Martín Melero, Javier Alcaraz & Mercedes Landete (15 Dec 2025): Improving the ranking position from the linear ordering problem with a limited budget, Journal of the Operational Research Society, DOI: 10.1080/01605682.2025.2589321

To link to this article: https://doi.org/10.1080/01605682.2025.2589321

© 2025 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group

Published online: 15 Dec 2025.

Submit your article to this journal

Article views: 448

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=tjor20

JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY
https://doi.org/10.1080/01605682.2025.2589321

RESEARCH ARTICLE

OPEN ACCESS Check for updates

Improving the ranking position from the linear ordering problem with a limited budget

Iñigo Martín Melero, Javier Alcaraz and Mercedes Landete

University Miguel Hernandez, Elche, Spain

ABSTRACT

Rankings establish comparative relationships among a set of items, such as athletes, companies, or universities. A common approach to synthesising multiple individual rankings into a single consensus ranking is through the Linear Ordering Problem (LOP). Traditionally, the LOP assumes that the input rankings are fixed; however, in practice, entities with access to financial resources may seek to strategically improve their position in the resulting aggregate ranking. This study introduces the *Improvement Budgeted Linear Ordering Problem* (IB-LOP), which addresses the scenario where a single item or client allocates a limited budget to enhance its standings across the criteria underlying the input rankings, aiming to achieve the highest possible position in the unified LOP-derived ranking.

We develop a non-linear bilevel optimisation framework to model the IB-LOP, which we subsequently reformulate into an equivalent linear single-level problem for tractability. Furthermore, we propose two extended formulations: one that maximises the LOP objective function and another that minimises the cost incurred, conditional on a target improvement in ranking. A heuristic algorithm is also presented to efficiently address larger instances of the problem. Extensive computational experiments, including a case study on hospital rankings, validate the effectiveness of the proposed models and heuristic. Key managerial insights derived from this work include the non-trivial impact of investment allocation across criteria and the observation that being the best in a single criterion is not a prerequisite for achieving a top position in the final ranking.

ARTICLE HISTORY
Received 8 May 2025
Accepted 8 November 2025

KEYWORDS
Linear ordering problem; upgrading; exact and heuristic techniques; budgeting

1. Introduction

Rankings, defined as the positioning of elements within a hierarchy or scale, are widely used to assess and compare the performance of entities in a shared context. This is evidenced in the academic literature across multiple domains. For instance, university rankings serve as a decision-making tool for various stakeholders in higher education, including prospective students, academic staff, employers, and policy-makers (Johnes, 2018). Several studies suggest that university rankings capture institutional excellence (Hammarfelt et al., 2017; Taylor & Braddock, 2007), showing strong correlations across different ranking systems (Olcay & Bulu, 2017), and can be instrumental for education policy reforms (Dakowska, 2013). Conversely, other works critically evaluate the methodologies behind these rankings and expose their limitations (Gadd, 2020; Goglio, 2016; Tofallis, 2012).

Beyond education, rankings are extensively used in sports (Hochbaum, 2006; Morales et al., 2016; Motegi & Masuda, 2012), academic publishing (Bornmann et al., 2018; Serenko & Bontis, 2009), and business and finance. In the latter context, rankings facilitate the assessment of risk exposure in firms and institutions (Kirchler et al., 2018; Trussel & Patrick, 2018; Tsai & Wang, 2013), and are also central to measuring corporate reputation, particularly within the Environmental, Social, and Governance (ESG) framework (Chen et al., 2007; Fombrun, 1998; Tischer & Hildebrandt, 2014). Recent contributions have further explored ESG-related rankings in the context of corporate sustainability and responsibility (Diez-Cañamero et al., 2020; Opferkuch et al., 2022; Patara & Dhalla, 2022).

Mathematical formulations are among the predominant tools for analysing and synthesising rankings. Examples include rank aggregation in cyclic sequences (García-Nové et al., 2017), centre ranking, and target visitation problems (Alcaraz et al., 2022). One of the most established models is the Linear Ordering Problem (LOP), initially introduced in Chenery and Watanabe (1958), which generates an optimal global ordering of elements by aggregating multiple pairwise preferences (Ceberio et al., 2015). The LOP is a well-known NP-hard combinatorial

CONTACT Mercedes Landete landete@umh.es University Miguel Hernandez, Elche 03202, Spain.
This article was originally published with errors, which have now been corrected in the online version. Please see Correction (http://dx.doi.org/10.1080/01605682.2026.2618330)

© 2025 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

optimisation problem with a wide range of applications, such as data rankability assessment (Cameron et al., 2021; Szczecinski & Sukheja, 2023). Due to its computational complexity, numerous metaheuristics have been proposed to efficiently solve large-scale LOP instances (Ali, 2024; Sakuraba et al., 2015). Apart from the LOP, there are other ways of aggregating rankings, such as the Rank Aggregation Problem, and even more recent proposals such as the model in Duan et al. (2025).

Several extensions of the LOP have been developed to address real-world problems, as a ranking can represent various applications such as routeing cities, sequencing tasks in production, or ordering clients. In this study, we focus on the latter scenario and refer to elements as *clients*, without loss of generality. Extensions of the LOP generally assume that rankings are static; however, in practical scenarios, clients may possess the capacity to improve their position within individual criteria, thus influencing the resulting LOP-derived ranking. Such improvements incur financial costs, which may vary depending on the criterion and magnitude of change. Given a limited budget, a client faces a resource allocation problem: how to distribute their investment to maximise their final position in the global ranking. To our knowledge, the literature lacks models that integrate budget constraints and dynamic, client-driven modifications of rankings.

In this paper, we introduce the Improvement Budgeted Linear Ordering Problem (IB-LOP), a novel optimisation framework that determines the most effective allocation of a budget across ranking criteria for a given client, with the goal of maximising its final rank. We conceptualise this as an intelligent decision-support system for strategic position enhancement in ranked systems. Proposed IB-LOP counts with many real-world practical applications, requiring as inputs a set of rankings, a client seeking to improve, their limited budget, and the cost to improve between each specific position of the set of rankings. Estimating these individual costs is the most challenging, involving extensive sector experience.

In business, magazines like *Forbes* are regularly ranking companies as Best Employers across different dimensions: net zero commitment, corporate diversity or specific employee segments like veterans, new graduates or women (Statista, 2025). Holding, and strategically improving positions in these rankings results very relevant, as it is associated with cumulative and interactive effects that add value to business portfolios (Filbeck et al., 2013), as well as enhancing talent attraction (Butler et al., 2016). Similarly, *Merco* ranks companies from mostly Latin-American countries in dimensions like corporate reputation and ESG compliance (Merco, 2024b). Precisely, Fernández Sánchez et al. (2012) suggest that in times of financial turmoil, being reputable and appearing in these rankings turns into a competitive advantage with a protective effect on the firm.

In higher education, universities that rank higher generally attract more and higher quality applications (Horstschräer, 2012; Luca & Smith, 2013), making ranking improvements relevant. For instance, the University of Pretoria (South Africa) and the Southern Cross University (Australia) gained positions in the *Quacquarelli Symonds* rankings (QS, 2025b) by strategically strengthening their position with regards to the UN Sustainable Development Goals (Southern Cross University, 2024; University of Pretoria, 2023). Others, like Nanyang Technological University (Singapore), gained positions by improving their citation count and faculty/student ratio (QS, 2025a). Ranking improvements are especially important in the case of business schools, where gaining positions in rankings like those from *Financial Times* (FT, 2025) allow for up to even doubling applications and the charging of prestige premiums (Thomas & Peters, 2007). In the context of universities, the cost of improvements is especially heterogeneous. It is not the same cost to increase the number of ERASMUS students as it is to increase the number of Nobel Prize winners, even if both values are criteria in the ranking.

Ultimately, the IB-LOP can be applied where belonging or strategically improving in rankings can attract customers and revenue, in sectors ranging from hospitality (NPR, 2010) to mobile technology (Mahmood, 2020).

This paper is structured as follows. We begin by introducing the foundational LOP notation, including its heuristics and extended models, followed by a formal description of the IB-LOP and its managerial implications. Subsequently, we propose both an exact model and a heuristic solution method. We validate our approach through extensive computational experiments, including a practical, real-world application to hospital rankings. Finally, the paper concludes with a summary of findings and directions for future research.

2. Preliminaries

2.1. Exact model for the LOP

There are different models and mathematical formulations for the LOP, like Chanas and Kobylański (1996); Schiavinotto and Stützle (2004); the binary linear one described in Martí and Reinelt (2011) is followed in this paper. Taking $V = \{1, \ldots, n\}$ as a set of clients and m as a number of given criteria $C_1, \ldots, C_m$ that rank those clients in V, there is a total of $n!$ different ranks of the clients in V.

The preference matrix D is defined as the square $n \times n$ matrix that indicates the number of times one client $i \in V$ outperforms another client $j \in V$ throughout all criteria, where element in row i and column j, d_{ij}, indicates the number of times that client i is ranked before j in the set of criteria. Given that ties are not allowed in the different rankings, for a given rank, when client i does not go before j it necessarily means that j is outperforming i, hence $d_{ij} + d_{ji} = m$. The goal of the LOP is to obtain a complete ranking of all clients in V that best represent the data, finding a best agreement among the different m criteria.

For all $i,j \in V$, the Z_{ij} binary variable is defined as follows,

$$Z_{ij} = \begin{cases} 1 & \text{if } i \text{ is ranked before } j \\ 0 & \text{otherwise} \end{cases}$$

The LOP can be effectively modelled as the linear problem expressed as follows.

$$\max \quad v(LOP) = \sum_{i \in V} \sum_{j \in V: j \neq i} d_{ij} Z_{ij} \tag{1}$$

$$\text{s.t.} \quad Z_{ij} + Z_{ji} = 1 \quad \forall i,j \in V : i < j \tag{2}$$

$$Z_{ij} + Z_{jk} + Z_{ki} \leq 2 \quad \forall i,j,k \in V : i \neq j, j \neq k, i \neq k \tag{3}$$

$$Z_{ij} \in \{0,1\} \quad \forall i,j \in V : i \neq j \tag{4}$$

The objective function (1) expresses the sum of the upper diagonal values in the ranked preference matrix, representing the total agreement of the ranking. The first family of constraints (2) states that either client i goes before client j or client j goes before client i. The second condition (3) rules out the existence of forbidden precedences; if i goes before j and j goes before k, then k cannot go before i. The variable z_{ij} is constrained to the binary domain in (4). The LOP does not consider the diagonal of matrix $D = (d_{ij})$, as it would represent the comparison of one client with itself, nor the lower diagonal elements. To rule out these positions of D from the model, $i \neq j$ applies everywhere. The NP-hard nature of the LOP makes it challenging to solve up to optimality as the instance size grows, hence the need for developing heuristics and metaheuristics (Martí & Reinelt, 2011).

2.2. Heuristics for the LOP

The most recent advances on heuristics for the LOP are compiled in Martí and Reinelt (2022). In addition, Charon and Hudry (2010) present an overview of conjectures and open problems related to the LOP, including a review of some heuristics. The first heuristic algorithms for the LOP were construction methods that obtained a linear ordering from scratch by adding one node at each step, iteratively. Among these, Chenery and Watanabe (1958) suggested initial ideas of ranking building from an input-output table, and Aujac (1960) introduced output coefficients to find linear orderings with few contradictions. Later, Becker (1967) proposed two methods: one based on special quotients for the ranking of sectors, and another based on rotations. Martí et al. (2012) suggest insertion as a construction method, where an arbitrary unassigned object is inserted into the partial solution at the best possible position. Improvement heuristics, which iteratively improve a solution by performing local changes, are also studied for the LOP. Laguna et al. (1999) suggest changing the positions of certain nodes and studying its effect in the objective function. Martí et al. (2012) also develop three heuristics for the LOP. One of them chooses a window and enumerates all possible orderings, *via* local enumeration. The other two apply compound movements as developed by Kernighan and Lin (1970) for graph partitioning: one is based on interchanging objects and another one builds upon insertions. Other heuristic improvement approaches include Lagrangian relaxation frameworks, as proposed by Belloni and Lucena (2003).

As for metaheuristics, multi-start procedures were employed by Chanas and Kobylański (1996), who exploited the symmetry property of the LOP to escape local optimality, and Martí et al. (2012), which were based again in Kernighan and Lin (1970). Scatter search was applied to the LOP by Campos et al. (2001), employing Greedy Randomised Adaptive Search Procedure (GRASP) as some of the diversification generation methods. Several tabu search algorithms were proposed by Laguna et al. (1999), including additional elements like path relinking or long term diversification. Garcia et al. (2006) combined Variable Neighbourhood Search (VNS) with short-term tabu search for improved outcomes, and Huang and Lim (2003) designed a hybrid genetic algorithm for the LOP. A memetic algorithm and an iterated local search were designed for the LOP by Schiavinotto and Stützle (2004). Garcia et al. (2019) also applied iterated local search, improving the solution quality by exploiting some properties of the LOP. A more recent work considers instance decomposition based on the Fourier transform for solving the LOP (Benavides et al., 2025).

2.3. Observations and examples

The application of the conventional LOP in data rankability has been widely analysed in literature. Example 1 illustrates a typical LOP applied to rankability of clients based on certain criteria, where the clients are ordered from best to worst in each individual criteria. This is the typical kind of problem

that is solved throughout the paper, varying the number of clients and criteria.

Example 1. *Clients A, B and C are ranked according to 8 criteria as follows,*

C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
A	A	A	A	A	B	B	C
B	B	B	B	B	A	C	B
C	C	C	C	C	C	A	A

Client A outperforms B and C in a total of 5 and 6 ranks, respectively. Client B is better than C in 7 occasions. The values in the lower diagonal, when B and C outperform A and when C is better than B, are obtained reciprocally. The preference matrix for the rank ABC would be as follows,

$$\begin{pmatrix} - & 5 & 6 \\ 3 & - & 7 \\ 2 & 1 & - \end{pmatrix}$$

This matrix coincides with the optimum matrix that bears the highest sum of the upper diagonal. The best rank is therefore ABC, with a $v(LOP)$ of 18. This case is quite intuitive, as A was the first one already in 5 criteria.

However, if B improves 1 place in criteria C_1, C_2 and C_3, it would become the best in those criteria. The preference matrix for the rank ABC would change to,

$$\begin{pmatrix} - & 2 & 6 \\ 6 & - & 7 \\ 2 & 1 & - \end{pmatrix}$$

However, the preference matrix for the rank BAC would have a larger sum of the upper diagonal,

$$\begin{pmatrix} - & 6 & 7 \\ 2 & - & 6 \\ 1 & 2 & - \end{pmatrix}$$

So, the best rank would change to BAC, with a $v(LOP)$ of 19.

In addition, one LOP can present several alternative optimal solutions. So, different ranking orders can have a same LOP value that might happen to be the optimal one, adding a fairness component to the problem (Anderson et al., 2022). In Example 2, different ranks are alternative optima, having the same $v(LOP)$.

Example 2. *Clients A, B, C, D and E are ranked according to 6 criteria as follows,*

C_1	C_2	C_3	C_4	C_5	C_6
A	A	D	D	E	C
C	B	C	E	D	A
D	E	E	B	A	B
E	D	A	C	B	D
B	C	B	A	C	E

The preference matrix for the rank ABCDE would be as follows,

$$\begin{pmatrix} - & 5 & 3 & 3 & 3 \\ 1 & - & 3 & 2 & 2 \\ 3 & 3 & - & 2 & 3 \\ 3 & 4 & 4 & - & 4 \\ 3 & 4 & 3 & 2 & - \end{pmatrix}$$

In this case, the $v(LOP)$ of 36 can be obtained both with rank ADECB and DAECB.

2.4. Extensions of the LOP

The LOP has been revisited by a number of authors, repurposing it for specific applications. For instance, Alcaraz et al. (2020) introduce an exact model for the clustered version of the LOP, where items belong to groups (e.g., teams or political parties), and a representative is selected and ranked per cluster. They also design a hybrid metaheuristic combining elements from scatter search and genetic algorithms, including an initial population set, adapted path relinking, insertion, mutation and interchange. Aparicio et al. (2020) define the exact model for the Kendall-τ partition ranking, for deriving a linear order among subsets rather than individual elements. Their work includes an empirical application on the Programme for International Student Assessment (PISA) rankings. In the work Labbé et al. (2023), it is argued that it may be possible for several set orders to be consistent with the same item order, and vice versa, that several item orders may be consistent with the same set order. In this work, the authors propose a bi-level model to compute an appropriate order of items when a set order is available, and another bi-level model to compute an appropriate order of sets when an item order is available. Both problems are formulated as bi-level optimisation problems and are illustrated using data from the Association of Tennis Professionals (ATP) dataset. Nowadays, sports rankings receive considerable attention. For example, in Vaziri et al. (2018), a set of properties for sport rankings is proposed such that, if satisfied, the ranking methods can be regarded as fair and comprehensive.

In another work, Hautz et al. (2020) propose the exact model for the weighted LOP, accounting for nodes with individual importance weights. They also designed a VNS algorithm for the weighted LOP, composed of the typical shaking, improving and updating steps. Their local search method was a descent procedure within the 2-neighbourhood. Vieira (2024) presents the exact model for the LOP with weighted rankings, which integrates both pairwise and positional preferences, and is motivated by electoral systems like instant-runoff voting. An empirical application to the first semi-final of Eurovision Song Contest 2021 is explained, and two heuristics were designed for this LOP extension: one iterated local search with 7 interchanges, and a memetic algorithm. Bertacco et al. (2008) develop the LOP with cumulative costs, modelling position changes with associated client-specific cost, proposing as exact models both a mixed integer model and an enumerative algorithm. They designed a dynamic programming heuristic that outperformed a reference GRASP algorithm.

All these extensions assume that rankings are static, and do not model dynamic, client-driven modifications of rankings according to budget constraints, hence the novelty of our work.

3. The IB-LOP statement

The main idea behind the IB-LOP is considering that one of the clients in V seeks to maximise the number of positions that it can improve in the given ranking when solving the LOP. The considered client $f \in V$ knows, can calculate or estimate beforehand the cost of improving one or more positions in each one of the m rankings. To carry out those improvements, it has a given limited budget. In addition, the upgrading is performed solely for client f, so other clients do not cooperate for the improvement.

To completely define a IB-LOP, different elements of the problem are required. First of all, m rankings of n clients, one per criteria, just as the ones employed for the base LOP. Then, the available budget B that the targeted client f is willing to invest for improving his LOP position and a matrix table indicating the improvement costs of the targeted client f. To individually define the improvements per position and criteria, we represent with H_{st} the cost of upgrading the targeted client f from position t to position $t-1$ in the ranking derived from criteria C_s, for $s = 1...m$ and $t = 2...P_s$, where P_s is the position occupied by targeted client f in criterion C_s, and $1 \leq P_s \leq n$.

Therefore, four main components are needed for the IB-LOP. First, a table containing m rankings of n clients, with a targeted client f that wants to improve their position in the global rank as determined by the LOP. From this table, the positions that client f occupies in each one of the m criteria is extracted. Targeted client f must provide an available budget B to invest in improving its position in the individual m criteria. Targeted client f must also estimate how much it would cost him to improve in each position until becoming the first one for the m criteria; if an improvement cannot be assumed by client f, it is modelled as having a cost greater than B.

The four necessary components for the IB-LOP are summarised as follows.

A first example where general notation and variables of the IB-LOP are illustrated is Example 3.

Table of m criteria and n clients

	C_1	C_2	C_3	C_4	…	C_m
Position 1					…	f
Position 2		f			…	
Position 3	f				…	
Position 4			f		…	
⋮	⋮	⋮	⋮	⋮	⋱	⋮
Position n				f	…	

Positions of client f: $P_1 = 3, P_2 = 2, P_3 = 4, P_4 = n, ..., P_n = 1$

Budget of client f = B.

Cost table of client f for the m criteria

	C_1	C_2	C_3	C_4	…	C_m
Position 1	–	–	–	–	…	–
Position 2	H_{12}	H_{22}	H_{32}	H_{42}	…	–
Position 3	H_{13}	–	H_{33}	H_{43}	…	–
Position 4	–	–	H_{34}	H_{44}	…	–
⋮	⋮	⋮	⋮	⋮	⋱	⋮
Position n	–	–	–	H_{4n}	…	–

Example 3. *Clients A, B and C are ranked according to 5 criteria as follows,*

C_1	C_2	C_3	C_4	C_5
C	B	C	B	B
B	C	A	A	A
A	A	B	C	C

The $v(LOP)$ is 10, obtained with the order BCA. One of the clients, in this case the worst of them, client A, has estimated how much of the budget it would take him to improve positions, with the following costs,

$$\begin{pmatrix} 6 & 2 & 7 & 8 & 9 \\ 5 & 8 & – & – & – \end{pmatrix}$$

So, improving 1 position and surpassing B in C_1 costs $H_{13} = 5$ units to A. Improving yet another position and becoming the best in C_1 costs an additional $H_{12} = 6$ units to A. Client A is already the second one in C_3, C_4 and C_5, so it can only improve from the second to first position in these criteria.

The minimum budget that A can employ to improve in any criteria is $H_{13} = 5$. Even though

there is an improvement costing less, $H_{22} = 2$, A should first pay $H_{23} = 8$ in order to unlock it. If a budget of 5 is considered and spent in H_{13}, thus making client A advance client B in C_1, the LOP solution of the resulting table of rankings would still be rank BCA. This means that the improvement has been done in vain, client A is still the last one.

However, with a budget of 7, improvement $H_{32} = 7$ is enabled. If A chooses to undergo this improvement instead of H_{13} and having 2 units remaining, the optimal order is altered to BAC with a $v(LOP)$ of 10, so A advances one position in the global rank.

When A invests a budget of 11, other solutions different from H_{32} lead to advancing one position in the global rank, like spending in $H_{23} = 8$, or in the combination of $H_{13} = 5$ and $H_{12} = 6$. Client A is capable of outperforming B and C when the spent budget is 13, by spending $H_{13} = 5$ and $H_{42} = 8$.

4. Managerial insights of the IB-LOP

When simulating further cases of the IB-LOP, several patterns suggest some properties of the problem. From the managerial perspective, understanding these properties is key for making the most appropriate decision on which improvements to invest in. In this section, we list some notable properties of the IB-LOP, stating each property with a Remark and proving it with an example.

To start, in most cases, improving the client implies reducing the value of the LOP. This means that, to a certain extent, improving the client means going against the "natural" ranking order, hence most changes of position in the client will be at the expense of less alignment in the criteria. We formalise this in Remark 4.1, which is illustrated in Example 4.

Remark 4.1. *Improving the position of a specific client may imply worsening the LOP optimal value.*

Example 4. *Clients A, B, C and D are ranked by 5 criteria as follows,*

C_1	C_2	C_3	C_4	C_5
C	C	A	B	C
B	A	C	D	A
D	B	D	C	D
A	D	B	A	B

Client D counts with a budget of 1 unit and has estimated a unitary cost matrix where all his improvements possible cost 1 unit. The preference matrix for the rank ABCD would be as follows,

$$\begin{pmatrix} - & 3 & 1 & 3 \\ 2 & - & 1 & 3 \\ 4 & 4 & - & 4 \\ 2 & 2 & 1 & - \end{pmatrix}$$

The optimal LOP solution is CABD with an objective value of 21. When improving 1 position for client D in C_3, the LOP solution remains CABD with an objective value of 20. Improving 1 position for client D in C_1, C_2 and C_4 has a LOP solution of CADB with objective values of 21, and improving that 1 position for client D in C_5 has a LOP solution of CDAB with an objective value of 20.

Another relevant observation is that, once a specific number of positions are improved by the client, it always becomes the best one in the ranking. This number of positions improved to be the best one in the ranking is generally lower than the maximum number of positions that the client can improve across all criteria (i.e., becoming the first in all of them). This means that the client can still be the best one in the ranking while not being the best client in all criteria, thus saving budget in unnecessary improvements. This is stated in Remark 4.2 and shown in Example 5.

Remark 4.2. *A client can be the best in a ranking without necessarily being the best for certain criteria.*

Example 5. *Clients A, B, C and D are ranked by 5 criteria as follows,*

C_1	C_2	C_3	C_4	C_5
A	A	B	B	C
C	D	D	D	A
D	B	C	C	B
B	C	A	A	D

Client D has estimated a unitary cost matrix where all his improvements possible cost 1 unit. The preference matrix for the rank ABCD would be as follows,

$$\begin{pmatrix} - & 3 & 2 & 3 \\ 2 & - & 3 & 3 \\ 3 & 2 & - & 2 \\ 2 & 2 & 3 & - \end{pmatrix}$$

The LOP solution is ABDC with an objective value of 17. The maximum improvement achievable for client D is given by the improvement vector [2,1,1,1,3] across criterias C_1 – C_5, requiring a budget of 8. However, client D is the best client in the ranking, with a LOP objective value of 20, for all possible improvement combinations that require a budget of only 5.

The distribution of the improvements throughout the criteria is important and problem-specific; some improvements are more effective than others. An improvement in a given criterion might only alter the LOP optimal value, without changing the LOP

order of clients. In this case, the LOP optimal value can decrease when the improvement goes against the "natural order" of the ranking, as stated before, or increase when it supports the "natural order" of the ranking (i.e., surpassing clients that were already behind the targeted client in the LOP solution). An improvement in a different criterion might change the LOP order of clients to the benefit of the targeted client, which is more budget effective. This is illustrated in Remark 4.3 and Example 6.

Remark 4.3. *The criteria chosen to invest in influence the resulting position in the ranking.*

Example 6. *Clients A, B, C, D, E, F, G and H are ranked by 7 criteria as follows,*

C_1	C_2	C_3	C_4	C_5	C_6	C_7
D	B	C	H	G	G	H
C	F	F	E	E	B	A
E	D	G	C	H	H	B
B	H	H	G	C	A	E
F	A	D	B	F	F	F
H	G	A	F	A	C	C
A	C	B	D	D	D	D
G	E	E	A	B	E	G

Client D has estimated a unitary cost matrix where all his improvements possible cost 1 unit. The preference matrix for the rank ABCDEFGH would be as follows,

$$\begin{pmatrix} - & 3 & 3 & 3 & 4 & 2 & 3 & 0 \\ 4 & - & 3 & 4 & 4 & 5 & 3 & 3 \\ 4 & 4 & - & 5 & 4 & 4 & 4 & 2 \\ 4 & 3 & 2 & - & 4 & 1 & 3 & 2 \\ 3 & 3 & 3 & 3 & - & 4 & 3 & 2 \\ 5 & 2 & 3 & 6 & 3 & - & 4 & 3 \\ 4 & 4 & 3 & 4 & 4 & 3 & - & 3 \\ 7 & 4 & 5 & 5 & 5 & 4 & 4 & - \end{pmatrix}$$

The LOP solution is HCGBFDAE with an objective value of 121. Improving only 1 position in any criteria does not alter the LOP solution. Improving that position in C_2, C_3, C_4, C_6, C_7 decreases the LOP objective value to 120, as it means surpassing clients that are better than client D in the LOP solution, going against its "natural order." Improving that unit in C_5 increases the objective value to 122, supporting the "natural order" of the ranking, as client D was already better than client A in the LOP solution. Likewise, becoming the first one in the ranking can require a budget as high as 20 (when applying an improvement vector (investment) of [0,0,0,5,4,6,5], for instance) and as low as 8 (when strategically applying, for example, an improvement vector (investment) of [0,0,3,0,0,5,0]), solely depending on the distribution of improvements across all criteria.

Finally, the several alternative optimal solutions of the LOP also play a role in determining the best attainable position for targeted client *f*. With respect to the best rank of the table at the starting situation, in some problems, solutions with targeted client *f* in better positions at the improved table might have the relative order between the other elements modified, to the benefit of client *f*. This is stated in Remark 4.4 and seen in Example 7.

Remark 4.4. *When improving the targeted client, the order between non-targeted clients can change.*

Example 7. *Clients A, B, C, D, E and F are ranked by 5 criteria as follows,*

C_1	C_2	C_3	C_4	C_5
A	F	E	F	C
F	A	F	E	F
C	D	B	B	A
D	B	D	D	B
E	E	C	C	E
B	C	A	A	D

Client D counts with a budget so that the only attainable investments is a 3-position improvement in C_1 and a 1-position improvement in C_2. The preference matrix for the rank ABCDEF would be as follows,

$$\begin{pmatrix} - & 3 & 2 & 3 & 3 & 1 \\ 2 & - & 3 & 3 & 2 & 0 \\ 3 & 2 & - & 2 & 2 & 1 \\ 2 & 2 & 3 & - & 2 & 0 \\ 2 & 3 & 3 & 3 & - & 1 \\ 4 & 5 & 4 & 5 & 4 & - \end{pmatrix}$$

The LOP solution is FAEBDC with an objective value of 51. This solution is unique. When the improvements are applied to client D in C_1 and C_2, the new LOP solution is FEBDCA with objective value of 51, and is also unique. Targeted client D climbs a step in the best rank and sends non-targeted client A to the last position.

These managerial insights indicate that there is a proven optimisation problem behind the IB-LOP, where the aim is maximising the effect of the improvements on the final LOP solution accounting for the limit in allowance. So, proposing a model that outputs the maximum amount of improvements achievable by the client inside its budget is justified.

5. An exact model for the IB-LOP

5.1. Complete mathematical model

To build the mathematical exact model for the IB-LOP, some further notation must be introduced.

- The binary parameter θ_{stj} is 1 if, in the ranking derived from criterion C_s, element f advances

element j when it is upgraded from position t to position $t-1$, and 0 in other case, for $s = 1 \ldots m$, $t = 2 \ldots P_s$, $j \in V$, $j \neq f$.

To solve the IB-LOP, we propose a bilevel linear program employing the following three families of variables, some of which are similar from those of the base LOP.

$$Z_{ij} = \begin{cases} 1 & \text{if, in the improved ranking, } i \text{ goes before } j \\ 0 & \text{otherwise} \end{cases} \quad \forall i, j \in V : i \neq j;$$

$$Y_{st} = \begin{cases} 1 & \text{if, in the ranking derived from criterion } C_s, \\ & \text{element } f \text{ moves from position } t \text{ to position } t-1 \\ 0 & \text{otherwise} \end{cases} \quad \forall s \in \{1, \ldots, m\}, t \in \{2, \ldots, P_s\};$$

d'_{ij}= value in row i and column j of the improved preference matrix D', for all $i, j \in V : i \neq j$.

The Y-variables take the value 1 whenever an improvement in the position is decided for a given criterion. For example, if the position in criterion s changes from 3 to 2, then $Y_{s3} = 1$ and $Y_{s2} = 0$. If the position changes from 5 to 2, then $Y_{s5} = Y_{s4} = Y_{s3} = 1$, and $Y_{s2} = 0$. If element f moves to the first position under criterion s, then $Y_{s2} = 1$.

The model, with the considered element as f, is presented as follows.

$$\max \quad Z = \sum_{j \in V : j \neq f} Z_{fj} \tag{5}$$

$$\text{s.t.} \quad \sum_{s=1}^{m} \sum_{t=2}^{P_s} H_{st} Y_{st} \leq B \tag{6}$$

$$d'_{ij} = d_{ij} \quad \forall i, j \in V : i \neq f, j \neq f \tag{7}$$

$$d'_{fj} = d_{fj} + \sum_{s=1}^{m} \sum_{t=2}^{P_s} \theta_{stj} Y_{st} \quad \forall j \in V : j \neq f \tag{8}$$

$$d'_{jf} = m - d'_{fj} \quad \forall j \in V : j \neq f \tag{9}$$

$$Y_{st} \leq Y_{s,t+1} \quad s = 1 \ldots m, t = 2 \ldots P_s - 1 \tag{10}$$

$$Y_{st} \in \{0, 1\} \quad s = 1 \ldots m, t = 2 \ldots P_s \tag{11}$$

$$\max \quad \sum_{i \in V} \sum_{j \in V : j \neq i} d'_{ij} Z_{ij} \tag{12}$$

s.t.

$$Z_{ij} + Z_{ji} = 1 \quad \forall i, j \in V : i < j$$
$$Z_{ij} + Z_{jk} + Z_{ki} \leq 2 \quad \forall i, j, k \in V : i \neq j, j \neq k, i \neq k$$
$$Z_{ij} \in \{0, 1\} \quad \forall i, j \in V : i \neq j$$

The objective function (5) reflects the maximisation of number of elements who fall behind f, in the $v(LOP)$. Constraint (6) ensures that all improvements fall inside budget B. In the improved preference matrix, (7) ensures that all elements not referring to f are left unchanged. The change in the element when $i = f$ is reflected in (8): the improvements that are indeed selected are added to the element. When $j = f$, the reciprocal operation to (8) is performed in (9). Constraint (10) ensures the correct order of improvements, driving the element in a criteria from its worst possible position to its best one through all the intermediate steps. The binary nature of Y_{st} is explicitly constrained in (11). The lower-level optimisation tasks correspond to the base LOP, considering the elements of the improved preference matrix d'_{ij} in (12) and inheriting the remaining constraints.

5.2. Simplified mathematical models

In general, the IB-LOP is a bilevel problem. We have a budget for improving the position of a client (there is a different cost for upgrading in each criterion) and we want to maximise the number of upgraded positions in the optimal solution for the LOP. Due to the complexity of the proposed model, it is first linearised and after transformed into a single level model, so as to be solved by standard MILP solvers.

5.2.1. Single level model

As the lower-level optimisation problem contains binary variables, standard techniques of bilevel programming problems cannot be applied. However, it is possible to rewrite the original bi-level model as a single level model if we use an exponential family of constraints. Specifically, if we define LOP as the set of all possible rankings of the n elements,

$$LOP = \{Z_{ij} \in \{0, 1\} : \ (2), \ (3)\},$$

then the lower-level objective function (5.1) can be replaced by the following constraint:

$$\sum_{i\in V}\sum_{j\in V:j\neq i} d'_{ij}Z_{ij} \geq \sum_{i\in V}\sum_{j\in V:j\neq i} d'_{ij}\overline{Z}_{ij} \quad \forall \overline{Z}_{ij} \in LOP.$$

So, the single-level exact model for the problem is:

$$\begin{aligned}
\max \quad & Z = \textstyle\sum_{j\in V:j\neq f} Z_{fj} \\
\text{s.t.} \quad & \sum_{s=1}^{m}\sum_{t=2}^{P_s} H_{st}Y_{st} \leq B \\
& d'_{ij} = d_{ij} \quad \forall i,j \in V : i \neq f, j \neq f \\
& d'_{fj} = d_{fj} + \sum_{s=1}^{m}\sum_{t=2}^{P_s}\theta_{stj}Y_{st} \quad \forall j \in V : j \neq f \\
& d'_{jf} = m - d'_{fj} \quad \forall j \in V : j \neq f \\
& Y_{st} \leq Y_{s,t+1} \quad s = 1...m, t = 2...P_s - 1 \\
& Y_{st} \in \{0,1\} \quad s = 1...m, t = 2...P_s \\
& \sum_{i\in V}\sum_{j\in V:j\neq i} d'_{ij}Z_{ij} \geq \sum_{i\in V}\sum_{j\in V:j\neq i} d'_{ij}\overline{Z}_{ij} \quad \forall \overline{Z}_{ij} \in LOP \\
& Z_{ij} + Z_{ji} = 1 \quad \forall i,j \in V : i < j \\
& Z_{ij} + Z_{jk} + Z_{ki} \leq 2 \quad \forall i,j,k \in V : i \neq j, j \neq k, i \neq k \\
& Z_{ij} \in \{0,1\} \quad \forall i,j \in V : i \neq j
\end{aligned}$$

Note that having an exponential family of constraints is common in combinatorial optimisation. For example, most hard problems, such as the Travelling Salesman Problem or the Vehicle Routeing Problem, use subtour elimination constraints, which form an exponential family.

5.2.2. Non-linear to linear

Some terms of the lower-level objective function are non-linear. So, some relevant changes would need to be done to the initial non-linear model. The lower-level objective function may be developed as follows.

$$\sum_{i\in V}\sum_{j\in V:j\neq i} d'_{ij}Z_{ij} = \sum_{i\in V:i\neq f}\sum_{j\in V:j\neq i,f} d_{ij}Z_{ij} + \sum_{j\in V:j\neq f} d'_{fj}Z_{fj} + \sum_{j\in V:j\neq f} (m - d'_{fj})Z_{jf}$$

Whilst the first term is linear, the second and third terms are non-linear and developed as follows, respectively.

$$\begin{aligned}
\sum_{j\in V:j\neq f} d'_{fj}Z_{fj} &= \sum_{j\in V:j\neq f}\left(d_{fj} + \sum_{s=1}^{m}\sum_{t=2}^{P_s}\theta_{stj}Y_{st}\right)Z_{jf} \\
&= \sum_{j\in V:j\neq f} d_{fj}Z_{fj} + \sum_{j\in V:j\neq f}\sum_{s=1}^{m}\sum_{t=2}^{P_s}\theta_{stj}Y_{st}Z_{fj} \\
\sum_{j\in V:j\neq f} (m - d'_{fj})Z_{jf} &= \sum_{j\in V:j\neq f} (m - d_{fj})Z_{jf} \\
&\quad - \sum_{j\in V:j\neq f}\sum_{s=1}^{m}\sum_{t=2}^{P_s}\theta_{stj}Y_{st}Z_{jf}
\end{aligned}$$

The products $Y_{st}Z_{jf}$ and $Y_{st}Z_{fj}$ represent the product of two binary variables, which is linearised with an additional variable W_{stij} such that the following constraints are added to the model.

$$\begin{aligned}
& W_{stij} \leq Y_{st} \quad \forall i,j \in V : i \neq j, s = 1...m, t = 2...P_s \\
& W_{stij} \leq Z_{ij} \quad \forall i,j \in V : i \neq j, s = 1...m, t = 2...P_s \\
& W_{stij} \geq Y_{st} + Z_{ij} - 1 \quad \forall i,j \in V : i \neq j, s = 1...m, t = 2...P_s
\end{aligned}$$

The final linear single level exact model is illustrated as follows.

$$\begin{aligned}
\max \quad & Z = \textstyle\sum_{j\in V:j\neq f} Z_{fj} \\
\text{s.t.} \quad & \sum_{s=1}^{m}\sum_{t=2}^{P_s} H_{st}Y_{st} \leq B \\
& d'_{ij} = d_{ij} \quad \forall i,j \in V : i \neq f, j \neq f \\
& d'_{fj} = d_{fj} + \sum_{s=1}^{m}\sum_{t=2}^{P_s}\theta_{stj}Y_{st} \quad \forall j \in V : j \neq f \\
& d'_{jf} = m - d'_{fj} \quad \forall j \in V : j \neq f \\
& Y_{st} \leq Y_{s,t+1} \quad s = 1...m, t = 2...P_s - 1 \\
& Y_{st} \in \{0,1\} \quad s = 1...m, t = 2...P_s \\
& \sum_{i\in V:i\neq f}\sum_{j\in V:j\neq i,f} d_{ij}Z_{ij} + \sum_{j\in V:j\neq f} d_{fj}Z_{fj} + \sum_{j\in V:j\neq f}\sum_{s=1}^{m}\sum_{t=2}^{P_s}\theta_{stj}W_{stfj} \\
& \quad + \sum_{j\in V:j\neq f} (m - d_{fj})Z_{jf} \\
& - \textstyle\sum_{j\in V:j\neq f}\sum_{s=1}^{m}\sum_{t=2}^{P_s}\theta_{stj}W_{stjf} \geq \sum_{i\in V}\sum_{j\in V:j\neq i} d'_{ij}\overline{Z}_{ij} \quad \forall \overline{Z}_{ij} \in LOP \\
& Z_{ij} + Z_{ji} = 1 \quad \forall i,j \in V : i < j \\
& Z_{ij} + Z_{jk} + Z_{ki} \leq 2 \quad \forall i,j,k \in V : i \neq j, j \neq k, i \neq k \\
& Z_{ij} \in \{0,1\} \quad \forall i,j \in V : i \neq j \\
& W_{stij} \leq Y_{st} \quad \forall i,j \in V : i \neq j, s = 1...m, t = 2...P_s \\
& W_{stij} \leq Z_{ij} \quad \forall i,j \in V : i \neq j, s = 1...m, t = 2...P_s \\
& W_{stij} \geq Y_{st} + Z_{ij} - 1 \quad \forall i,j \in V : i \neq j, s = 1...m, t = 2...P_s
\end{aligned} \tag{13}$$

5.3. Extended mathematical models

The problem under consideration, IB-LOP, generally admits a multitude of alternative solutions. That is, in many cases there are several ways of allocating the budget that lead to the same position in the ranking for client f. Moreover, the value of the LOP, $v(LOP)$, for each of these alternative solutions may potentially differ. Therefore, it makes sense to formulate additional optimisation problems that allow us to select, from the pool of optimal solutions, the one (or those) that optimise other criteria. Specifically, we propose solving two additional problems: one that maximises $v(LOP)$, and another that minimises the expenditure. So, while the IB-LOP exact model provides solutions where the improvement of client f is maximum, it does not optimise achieving Z with the maximum $v(LOP)$ or with the minimum cost. A good $v(LOP)$ is desirable because it represents the level of agreement in the global rank; achieving Z^* maximising the LOP objective leads to a final ranking where the total agreement of the individual rankings is maximised. In the same

line, a low cost is desirable because it allows for savings from the client's side; achieving Z^* by spending the least possible cost is relevant from an economic and resource perspective. Each one of these secondary objectives can be solved in 2 different steps. For the first step, the optimum amount of clients that client f outperforms is calculated with the exact model, noted as Z^*. Later, constraint (13) is excluded, and constraint

$$\sum_{j\in V:j\neq f} Z_{fj} = Z^* \tag{14}$$

is included in the model. To obtain the solution that achieves Z^* with minimum cost, IB-LOP-MinC, then in this second step objective function (15) is included, replacing objective function (5).

$$\min \quad \sum_{s=1}^{m}\sum_{t=2}^{P_s} H_{st}Y_{st}, \tag{15}$$

As the $v(LOP)$ represents the agreement between the criteria, client f might be also interested in having the highest possible agreement so as to prevent doubts about their position in the global rank, reputation-wise. Likewise, to obtain the solution that achieves Z^* with a maximum value of the $v(LOP)$, IB-LOP-MaxL, then in this second step objective function (16) is included, replacing objective function (5).

$$\begin{aligned}\max \quad & \sum_{i\in V:i\neq f}\sum_{j\in V:j\neq i,f} d_{ij}Z_{ij} + \sum_{j\in V:j\neq f} d_{fj}Z_{fj} \\ & + \sum_{j\in V:j\neq f}\sum_{s=1}^{m}\sum_{t=2}^{P_s}\theta_{stj}W_{stfj} + \sum_{j\in V:j\neq f}(m-d_{fj})Z_{jf} \\ & - \sum_{j\in V:j\neq f}\sum_{s=1}^{m}\sum_{t=2}^{P_s}\theta_{stj}W_{stjf}\end{aligned} \tag{16}$$

6. A Heuristic approach for the IB-LOP

The main objective of the IB-LOP is to distinguish which improvements a client must prioritise when improving in given criteria and having limited budget. As seen in previous sections, the exact model provides how many positions our targeted client f must improve in each individual criteria in order to optimise the number of customers it outperforms with a limited budget.

As with other MILP problems, exact models are useful up to their limits of computational effort, point after which the domain of heuristics and metaheuristics enters in game. The complexity of the IB-LOP is mainly due to constraint (13), where all permutations of a given LOP are of $O(n!)$ nature, one of the most time and memory consuming types. Another difficulty of the IB-LOP remains in the fact that optimising the required improvements at the preference matrix level is not enough, as they might not be feasible budget-wise according to the cost table nor position-wise according to the criteria table; all are deeply interrelated. In this work, deterministic heuristic IB-LOP-Perm has been self-tailored for the IB-LOP, based on observations made after solving many instances of the IB-LOP. These observations consisted in the fact that time resolution of the IB-LOP could effectively be reduced by selectively building constraint (13) only with certain LOP permutations, instead of the $n!$ total amount.

The procedure of the IB-LOP-Perm is represented in Algorithm 1. In short, it counts with 5 main steps: calculating the initial best order, creating an initial pool of LOP permutations, solving the reduced IB-LOP, checking the solution and iterating if the Z provided by the IB-LOP is different from the Z provided by the check, until when they match.

Algorithm 1 Procedure IB-LOP-Perm

1. Calculate best order $\rightarrow Z_{IB-LOP}, LOP_X$
2. Create permutations $\rightarrow LOP_{red} = \{LOP_1, LOP_2, \ldots, LOP_n\}$
3. **while** not (stopping criteria) **do**:
4. Solve IB-LOP $\rightarrow Z_{pred}, LOP_{pred}, IMP_{pred}$
5. Check solution $\rightarrow Z_{check}, LOP_{check}$
6. **if** Z_{pred} is different from Z_{check}:
7. Add $LOP_{check} \rightarrow LOP'_{red}$
8. **else:**
9. Activate stopping criteria
10. **end while**
11. **return** Z, LOP, IMP

The specific interrelation between the preference matrix and the criteria and cost tables prevents the application of standard LOP heuristics, in which all combinations are feasible. Consequently, Algorithm 1 represents a novel approach. Its only resemblance to existing metaheuristic algorithms lies in the iterative mechanism of adding permutations, which can be loosely compared, in a broad sense, to the position permutation strategy used in the heuristic of Laguna et al. (1999).

Each of the steps of Algorithm 1 are explained in more detail in the following subsections, and illustrated when solving Example 8.

Example 8. *Clients A, B, C, D, E, F and G are ranked by 5 criteria as follows,*

C_1	C_2	C_3	C_4	C_5
C	F	A	E	D
D	A	D	C	F
A	C	G	F	A
E	D	C	D	C
G	E	B	A	E
B	G	E	B	B
F	B	F	G	G

Of the available clients, Client F is chosen as targeted client f to perform improvements, having a budget of 20 and a cost table as follows,

$$\begin{pmatrix} 4 & - & 9 & 5 & 5 \\ 1 & - & 3 & 7 & - \\ 9 & - & 5 & - & - \\ 9 & - & 7 & - & - \\ 5 & - & 5 & - & - \\ 2 & - & 9 & - & - \end{pmatrix}$$

6.1. Calculation of best order

As a preliminary step towards building the initial reduced pool of permutations, the LOP of the base table of criteria is solved, without applying any improvements. This LOP is calculated through its exact model, Eqs. (1)–(4), suitable enough for medium size instances (larger instances of IB-LOP-Perm may require heuristics to solve the base LOP). In any case, the output order of this LOP, LOP_X, is then employed to construct the rest of the initial reduced pool of permutations. When more than one rank is optimal, sharing thus $v(LOP)$, the solution with highest (or more favourable) Z_{IB-LOP} for targeted client f is chosen among the others to ensure high quality of the pool of permutations. This process is represented in the continuation of Example 8.

Example 8 (**cont.**).

The preference matrix for the rank ABCDEFG would be as follows,

$$\begin{pmatrix} - & 5 & 3 & 2 & 4 & 2 & 5 \\ 0 & - & 0 & 0 & 1 & 2 & 2 \\ 2 & 5 & - & 3 & 4 & 3 & 4 \\ 3 & 5 & 2 & - & 4 & 3 & 5 \\ 1 & 4 & 1 & 1 & - & 3 & 4 \\ 3 & 3 & 2 & 2 & 2 & - & 3 \\ 0 & 3 & 1 & 0 & 1 & 2 & - \end{pmatrix}$$

In this case, four alternative ranks share the optimal of $v(LOP) = 77$: ACDEFGB, CDAEFGB, DACEFGB and CDFAEGB. While in the first three, for targeted client F, $Z_{IB-LOP} = 2$, in the last one $Z_{IB-LOP} = 4$. So, the last rank is selected, and thus $LOP_X = CDFAEGB$ with $Z_{IB-LOP} = 4$.

6.2. Creation of permutations

To construct the initial pool of permutations, LOP_X is employed as a reference. Targeted client f is then removed from LOP_X, resulting in $LOP_X \setminus \{f\}$. The initial pool of permutations, $LOP_{red} = \{LOP_1, LOP_2, \ldots, LOP_n\}$, is built by subsequently inserting client f in all possible positions of $LOP_X \setminus \{f\}$, keeping the relative order between the rest of non-targeted clients fixed. This is illustrated again in the following continuation of Example 8.

Example 8 (**cont.**).

For $LOP_X = CDFAEGB$, removing F results in $LOP_X \setminus F = CDAEGB$. The initial pool of permutations would result in,

$$LOP_{red} = \{FCDAEGB, CFDAEGB, CDFAEGB, CDAFEGB, CDAEFGB, CDAEGFB, CDAEGBF\}$$

6.3. IB-LOP resolution

As following step, the IB-LOP of the problem is solved employing $\overline{Z}_{ij} \in LOP_{red}$ instead of $\overline{Z}_{ij} \in LOP$, which reduces time and memory complexity from $O(n!)$ to $O(n)$. The optimal values of the Y variables obtained from LOP_{red} define an improvement vector, denoted as IMP_{pred}. This is illustrated with the continuation of Example 8.

Example 8 (**cont.**).

When adopting $\overline{Z}_{ij} \in LOP_{red}$, the resolution lasts for 1.9 s. The solution provided by the problem, for F, is summarised as follows,

$$IMP_{pred} = [0, 0, 1, 1, 0], Z_{pred} = 6,$$
$$LOP_{pred} = FACDEGB$$

6.4. Solution check

Solving the linear single-level optimisation model for IB-LOP using LOP_{red} instead of LOP is much less costly in terms of time and memory, but its solutions may not be feasible for the full model with all the constraints. In what follows, we analyse how to check if a solution is IB-LOP feasible, how to construct a feasible solution from it if it is not, and how to attempt to improve a feasible solution. The checking process consists in applying the vector IMP_{pred} to the base ranking table, solving its LOP, and obtaining its Z_{check} and LOP_{check} order for targeted client f. Again, the LOP_{check} order with highest Z_{check} for the targeted client f is selected in case of multiple optimal solutions. If correct, Z_{pred} and Z_{check} match, ending the IB-LOP-Perm and providing the final solution. As seen in the following

continuation of Example 8, this is not always the case, which requires some further operations.

Example 8 (**cont.**).

When applying $IMP_{pred} = [0, 0, 1, 1, 0]$ *to the base ranking table, the improved preference matrix for the rank ABCDEFG is as follows,*

$$\begin{pmatrix} - & 5 & 3 & 2 & 4 & 2 & 5 \\ 0 & - & 0 & 0 & 1 & 2 & 2 \\ 2 & 5 & - & 3 & 4 & 2 & 4 \\ 3 & 5 & 2 & - & 4 & 3 & 5 \\ 1 & 4 & 1 & 1 & - & 2 & 4 \\ 3 & 3 & 3 & 2 & 3 & - & 3 \\ 0 & 3 & 1 & 0 & 1 & 2 & - \end{pmatrix}$$

The IB-LOP obtained that this preference matrix had an optimal $LOP_{pred} = FACDEGB$, *which has a* $v(LOP) = 77$. *However, the IB-LOP on the reduced pool of permutations overlooked that best* $v(LOP) = 78$ *is provided instead with* $LOP_{check} = DFACEGB$. *So,* $Z_{pred} = 6$ *while* $Z_{check} = 5$, *requiring further operations.*

6.5. Iteration

When Z_{pred} and Z_{check} do not match, the IB-LOP has overlooked optimal orders worthy of being evaluated, due to the reduced (and limited) pool of permutations. To correct the results of the IB-LOP, the optimal order LOP_{check} that produces this mismatch is added to the initial pool of permutations, generating LOP'_{red}, and then IB-LOP and subsequent operations are run again. This process of addition of LOP_{check} is continued, iteratively enlarging the pool of permutations, until when Z_{pred} is the same as Z_{check}. By this moment, IB-LOP-Perm is considered to have finalised. This last step of the heuristic is followed in the continuation of Example 8.

Example 8 (**cont.**).

When adding $LOP_{check} = DFACEGB$ *to* LOP_{red}, LOP'_{red} *is obtained as follows,*

$$\begin{aligned} LOP'_{red} = \{&FCDAEGB, CFDAEGB, CDFAEGB, \\ &CDAFEGB, CDAEFGB, CDAEGFB, \\ &CDAEGBF, DFACEGB\} \end{aligned}$$

The resolution of the IB-LOP on LOP'_{red} *brings the following solutions,*

$$\begin{aligned} &IMP'_{pred} = [0, 0, 0, 1, 1], \quad Z'_{pred} = 6, \\ &LOP'_{pred} = FACDEGB \end{aligned}$$

Applying $IMP'_{pred} = [0, 0, 0, 1, 1]$ *to the base ranking table, the improved preference matrix for the rank ABCDEFG is as follows,*

$$\begin{pmatrix} - & 5 & 3 & 2 & 4 & 2 & 5 \\ 0 & - & 0 & 0 & 1 & 2 & 2 \\ 2 & 5 & - & 3 & 4 & 2 & 4 \\ 3 & 5 & 2 & - & 4 & 2 & 5 \\ 1 & 4 & 1 & 1 & - & 3 & 4 \\ 3 & 3 & 3 & 3 & 2 & - & 3 \\ 0 & 3 & 1 & 0 & 1 & 2 & - \end{pmatrix}$$

The optimal of this preference matrix is $v(LOP) = 77$, *for orders FACDEGB, FCDAEGB and FDACEGB. In all of them,* $Z'_{check} = 6$, *which matches* $Z'_{pred} = 6$, *so the IB-LOP-Perm is finished. This second iteration lasted for 2.3 s, resulting in a total 4.2 s of IB-LOP-Perm, as compared to the resolution of 36.8 s when considering* $\overline{Z}_{ij} \in LOP$. *For this problem, both achieve the optimal value of* $Z^* = 6$.

7. Computational experience

The computational experiments for this work have been performed on a HP Z4 G4 workstation computer, containing 6 physical cores and 12 logical processors of an Intel(R) Xeon(R) W-2235 CPU @ 3.80 GHz, and 64.0 GB of RAM. The programming language employed has been Python through the Pyomo framework (Bynum et al., 2021; Hart et al., 2011). Gurobi optimiser (Gurobi Optimisation, LLC, 2024), version 11.0.3, has been used as MILP solver.

The aim of these computational experiments is to visualise the behaviour and results of the proposed problem on different kinds of instances. This problem is new to literature and requires a table with criteria that ranks the clients, a specific targeted client f, its budget and its associated cost table as input data, in contrast to LOP instances that only require LOP matrices available in standard libraries like LOLIB for their experiments (Campos & Martí, 2024). So, representative instances for the problem have been created and employed for this work, and are available publicly in GitHub (Martin-Melero, 2024). A set of diverse instances is created by altering the two main dimensions of the problem: number of clients and number of criteria. While increasing the number of criteria moderately impacts the computational time, changing the number of clients directly impacts the size of the LOP preference matrix, significantly affecting the computational time. Therefore, three kinds of instances are defined in this work.

Experiments performed on instances with 2, 3 and 4 clients are considered small instances, as computation times for solving them in an exact way are negligible ($t < 5$ s). Instances with 5, 6, 7 and 8

clients are considered medium size instances, as their computation time is higher but remains manageable ($t < 20$ min). The exact model only works with up to 8 clients, failing with 9 clients due to a RAM memory demand that surpasses the available. So, experiments with 10, 20 and 30 clients are referred to as large instances because the exact model is not able to solve them. For all small, medium and large cases, simulated instances include five different levels of number of criteria: 7, 11, 14, 18 and 21 criteria.

In some of the experiments, Hexaly models are also developed for the IB-LOP. Hexaly, formerly known as LocalSolver, is a fast, scalable global optimisation software based on nonlinear and set modelling constructs. It employs not only MILP techniques, but also techniques like column generation or local search, not employed by traditional solvers (Hexaly, 2025). It has been applied to many problems inside location science and logistics, like route optimisation (Martí et al., 2024), workforce and production scheduling (Antuori et al., 2020) and supply network management (Ratli et al., 2022), among others. Some recent works like Colmenar et al. (2024); Hochbaum et al. (2025) employ it to compare and benchmark their developed heuristics.

For the small instances, experiments of the baseline IB-LOP and its extended variants minimising the cost (IB-LOP-MinC) and maximising the $v(LOP)$ (IB-LOP-MaxL) are carried out. At this level, elapsed time is not the main object of study, but rather the alternative solutions provided by the extended models. The medium instances have been solved with the exact IB-LOP model, the heuristic IB-LOP-Perm and Hexaly. The main point of these experiments is proving the adequacy and efficiency of our heuristic, in order to generalise it in the large instances. For the large instances, the heuristic is further benchmarked to prove its capabilities. Its best Z achieved is compared with two further methods. One of them produces random improvement vectors and calculates their Z, equivalent to a random search in the solution space. The other is a greedy technique consisting of making the cheapest improvement available until the budget runs out. These large instances are not only unattainable by the exact model, but also by Hexaly, which is also limited to maximum 8 clients due to surpassing the RAM memory demand when building the LOP permutations.

Therefore, the small instances comprise a total of 15 cases to be solved for the IB-LOP, IB-LOP-MinC and IB-LOP-MaxL. The medium instances comprise a total of 20 cases to be solved for the IB-LOP, the IB-LOP-Perm and Hexaly. The large instances comprise a total of 15 cases to be solved for the greedy, random search and IB-LOP-Perm. All cases are referred to by their number of clients and criterias e.g., case 5-18 contains 5 clients and 18 criteria.

7.1. Results on small instances

The experiments performed on the small instances are represented in Table 1. The first three columns refer to the characteristics of the instance: number of clients, number of criteria and available budget. The fourth column represents Z_{IB-LOP}, defined as the number of clients initially outperformed by client f without applying any improvements. If $z_{IB-LOP} = 0$, it is because f is the last in the LOP rank. The optimal Z^* given by the exact algorithms is represented on the fifth column. Next, three groups of two columns reflect the total cost incurred and the value of the LOP objective when considering the base IB-LOP, the cost-minimising IB-LOP-MinC and the LOP-maximising IB-LOP-MaxL.

The simplest cases to analyse are those containing only 2 clients. In these, client f is the second one and only has to outperform the first one, so the Z^* is 1 in all of them as far as there is enough budget to achieve it. Some relevant behaviour of the exact models already appears at this most basic level. For

Table 1. Solutions of the analysis on 2, 3 and 4 clients.

Problem definition					IB-LOP		IB-LOP-MinC		IB-LOP-MaxL	
Clients	Criteria	Budget	Z_{IB-LOP}	Z^*	Cost	$z(LOP)$	Cost	$z(LOP)$	Cost	$z(LOP)$
	7	9	0	1	6	4	4	4	9	5
	11	11	0	1	10	6	6	6	10	7
2	14	16	0	1	10	7	3	7	16	10
	18	12	0	1	9	9	9	9	9	9
	21	26	0	1	12	11	9	11	22	14
	7	17	0	2	17	12	6	11	14	13
	11	15	0	1	14	18	10	19	13	19
3	14	21	0	2	14	24	13	24	21	27
	18	13	0	2	8	27	6	27	13	29
	21	18	0	2	16	34	7	33	18	37
	7	15	2	3	4	30	3	30	13	33
	11	19	1	3	18	37	12	38	18	41
4	14	12	2	3	8	45	2	45	11	49
	18	9	0	2	8	62	7	62	9	63
	21	31	0	3	29	67	20	67	30	70

instance, the base IB-LOP of case 2-7 provides a solution that costs 6 units and obtains a $v(LOP)$ of 4. When minimising the cost once Z^* is known, however, a different solution with a lower cost of 4 units and same $v(LOP)$ is found. If a more solid agreement between criteria is sought (higher $v(LOP)$), applying the IB-LOP-MaxL finds a solution with a higher $v(LOP)$ of 5 at the expense of spending all the available budget. The base IB-LOP seems to output an intermediate solution cost-wise with respect to the IB-LOP-MaxL and IB-LOP-MinC. $v(LOP)$-wise, however, the solution obtained by the IB-LOP-MinC is the same as the IB-LOP but with less budget, so it is more desirable. Solutions where one same budget can produce different $v(LOP)$ s also appear. In case 2-11, both the IB-LOP and IB-LOP-MaxL solutions cost 10 units; however, the latter obtains a solution with better $v(LOP)$ than the former. If a cost of 10 units is invested, it would be desirable to improve on the criteria as provided by the IB-LOP-MaxL. Similarly as before, IB-LOP-MinC obtains a same $v(LOP)$ as the IB-LOP but with 40% less cost. In other cases, the base IB-LOP obtains the same solution as its variants. One example would be case 2-18 where, from an available budget of 12, all exact algorithms decide to employ 9 units and the solution has a $v(LOP)$ of 9.

In the 3 client cases, f is still the last one in the base rankings. On the other hand, the 4 client cases are much more diverse, featuring f in different positions of the base ranking: last (case 4-21), third (case 4-11) and second (case 4-7), as indicated by the current Z. Likewise, both 3 and 4 client cases are diverse in the positions improved by the solution: most times the Z^* corresponds to becoming the best client after the improvements (cases 3-18 and 4-11), but sometimes the budget only allows f to become the second (cases 3-11 and 4-18). In these experiments, the variants of the IB-LOP achieve larger differences in terms of cost and $v(LOP)$ with respect to the base case. Overall, the largest relative cost difference between the IB-LOP and the IB-LOP-MinC is of 70% while the largest $v(LOP)$ relative difference between the IB-LOP and the IB-LOP-MaxL is of 30%, both for case 2-14.

The distribution of the improvements among the different criteria of the small instances solutions is described in Table 2. There is a wide variety of distribution of improvements proven by the fact that, inside each case, there is not an exact repetition of distribution of improvements between the IB-LOP, IB-LOP-MinC and IB-LOP-MaxL. When analysing 2 clients, all improvements among the criteria are maximum one unit. At this level, cases with more criteria tend to distribute more their improvements: solutions in cases 2-18 and 2-21 improve in at least four criteria, while cases 2-11 and 2-14 in a minimum of two. In the solutions of cases 2-7 and 2-14, an improvement for a certain criteria is not repeated among IB-LOP, IB-LOP-MinC and IB-LOP-MaxL simultaneously. However, in the rest of cases, at least one criteria is chosen for improvement by the three models: C_9 for case 2-11 and C_9, C_{12}, C_{16} for cases 2-18 and 2-21. This highlights the fact that some specific improvements are relevant in order to change the global rank to the favour of client f.

The distribution of improvements among criteria in the cases with 3 clients follow some of the patterns of the cases with 2 clients. All cases except case 3-11 have at least one criteria improved in common between the IB-LOP, IB-LOP-MinC and IB-LOP-MaxL. In all cases, solutions from the IB-LOP-MaxL model distribute their improvements among the largest number of criteria, when comparing the three models. For instance, in cases 3-18 and 3-21, IB-LOP-MaxL distributes its improvements among 5 criteria, while IB-LOP and IB-LOP-MinB distribute their improvements among 3 criteria only. On the other hand, solutions provided by IB-LOP-MinB tend to do less amount of improvements, as the minimum cost is sought. For example, in case 3-7, while IB-LOP and IB-LOP-MaxL improve 4 and 5 positions respectively, IB-LOP-MinB only improves 3 positions. Nonetheless, these 3 positions are enough so as to change the global rank.

When analysing 4 clients, all cases have at least one criteria with improvements common for the

Table 2. Number of improvements per criteria in analysis on 2, 3 and 4 clients.

Clients	Criteria	IB-LOP	IB-LOP-MinC	IB-LOP-MaxL
2	7	C_4: 1	C_2: 1	C_1: 1, C_2: 1
	11	C_9: 1, C_{11}: 1	C_2: 1, C_9: 1	C_2: 1, C_8: 1, C_9: 1
	14	C_1: 1, C_{13}: 1	C_9: 1, C_{12}: 1	C_1: 1, C_2: 1, C_8: 1, C_9: 1, C_{12}: 1
	18	C_9: 1, C_{12}: 1, C_{16}: 1, C_{18}: 1	C_2: 1, C_9: 1, C_{12}: 1, C_{16}: 1	C_9: 1, C_{12}: 1, C_{16}: 1, C_{18}: 1
	21	C_9: 1, C_{12}: 1, C_{16}: 1, C_{21}: 1	C_2: 1, C_9: 1, C_{12}: 1, C_{16}: 1	C_1: 1, C_2: 1, C_8: 1, C_9: 1, C_{12}: 1, C_{16}: 1, C_{18}: 1
3	7	C_1: 2, C_3: 1, C_4: 1	C_1: 1, C_4: 2	C_1: 1, C_2: 1, C_3: 1, C_4: 2
	11	C_7: 1, C_8: 2	C_1: 1, C_3: 1, C_6: 1	C_1: 1, C_2: 1, C_3: 1
	14	C_1: 1, C_3: 2, C_8: 1, C_{12}: 1	C_1: 1, C_7: 1, C_8: 1, C_9: 1, C_{12}: 1	C_1: 1, C_3: 2, C_7: 1, C_8: 1, C_{10}: 2, C_{11}: 1
	18	C_{14}: 1, C_{16}: 1, C_{18}: 1	C_5: 1, C_{14}: 1, C_{18}: 1	C_2: 1, C_5: 1, C_{14}: 1, C_{15}: 1, C_{18}: 1
	21	C_3: 1, C_4: 1, C_{20}: 2	C_3: 1, C_{12}: 1, C_{20}: 1	C_3: 2, C_5: 1, C_8: 2, C_{12}: 1, C_{20}: 1
4	7	C_2: 1, C_3: 1	C_2: 1, C_4: 1	C_2: 2, C_3: 1, C_4: 2
	11	C_1: 1, C_3: 2, C_4: 1, C_{11}: 1	C_1: 1, C_3: 2, C_8: 1, C_{10}: 2	C_1: 1, C_3: 2, C_8: 3, C_9: 1, C_{10}: 2
	14	C_7: 1, C_{10}: 1	C_5: 1, C_{10}: 1	C_1: 1, C_5: 2, C_{10}: 3
	18	C_9: 1, C_{12}: 1, C_{14}: 1	C_9: 2, C_{12}: 1	C_9: 2, C_{12}: 1, C_{14}: 1
	21	C_{10}: 3, C_{12}: 1, C_{19}: 1, C_{20}: 3	C_1: 1, C_{12}: 1, C_{15}: 1, C_{18}: 1, C_{19}: 1, C_{20}: 3	C_1: 1, C_{11}: 2, C_{12}: 1, C_{15}: 3, C_{19}: 1, C_{20}: 3

three models. Moreover, the cases with 4 clients portray three main strategies when choosing how to distribute the improvements. One strategy is to distribute the improvements in as many criterias as possible: in case 4-21 for the IB-LOP-MinB, improvements are distributed among 6 criteria. The opposite strategy would be to improve more in a lower quantity of criteria: in case 4-14 for the IB-LOP-MaxL, only 3 criteria are chosen, but 3 and 2 improvements are done in 2 of the criteria. Case 4-21 for the IB-LOP and IB-LOP-MaxL would represent a mixed strategy, where improvements are focused on some criteria and also distributed to a lesser extent in other criteria.

7.2. Results on medium instances

In this analysis, we have only studied the IB-LOP, and we have not considered the secondary objectives, cost and LOP. The results of the experiments with 5, 6, 7 and 8 clients is shown in Table 3. The analysis focus is now comparing the exact IB-LOP with Hexaly and deterministic heuristic IB-LOP-Perm. The first five columns of the Table remain the same as in Table 1. The sixth column represents the time taken by the exact algorithm to find the optimal solution Z^*. The next column reflects the time taken by the Hexaly model to find optimal solution Z^*, while the last column illustrates the time taken by heuristic IB-LOP-Perm to reach it. Both Hexaly and IB-LOP-Perm succeed in always reaching Z^*.

The increase in computing complexity of the problem when it becomes larger is observed in this set of results. While the exact algorithm experiments for 5 clients take 1-3 s, the cases for 6 clients take 5-10 s. For 7 clients, this number climbs up to 0.5-1.5 min. When analysing a specific level of number of clients, the cases with more criteria in the ranking table take longer. This difference between criteria is also accentuated as the number of clients rises: whilst the difference is around 2 s between cases 5-7 and 5-21, it remains of 50 s between cases 7-7 and 7-21. This increase in the size of the problem also allows for a wider diversity in the problem formulation: in case 5-18 client f starts the last and improves only 2 positions, in case 5-14 it starts the second-to-last and improves until becoming the best one and in case 7-7 it also starts second-to-last and improves until becoming the second.

Table 3. Solutions of the analysis on 5, 6, 7 and 8 clients.

Problem definition					Time (s)		
Clients	Criteria	Budget	Z_{IB-LOP}	Z^*	IB-LOP	Hexaly	IB-LOP-Perm
	7	43	0	4	0.9	0.6	1.2
	11	33	0	3	1.9	1.4	1.4
5	14	15	1	4	1.9	0.9	1.6
	18	25	0	2	2.2	2.2	1.5
	21	20	2	4	2.4	2.6	1.7
	7	37	0	4	4.8	3.8	1.6
	11	38	0	5	6.2	2.8	1.7
6	14	20	1	5	7.3	3.3	1.7
	18	64	0	5	8.9	4.1	1.7
	21	9	2	5	10.4	5.6	1.9
	7	33	1	5	33.6	17.6	3,4
	11	55	1	6	48.5	20.3	1.8
7	14	15	0	4	58.3	28.1	2.2
	18	47	0	4	75.8	37.3	4,4
	21	20	2	5	86.2	42.2	2.3
	7	28	3	7	360.5	136.8	1.9
	11	69	0	7	509.4	220.5	3.0
8	14	176	0	7	640.1	249.2	2.5
	18	110	0	7	793.6	320.2	5.9
	21	24	2	7	817.5	364.1	6.4

In the analyses considering 5 clients, the exact IB-LOP, Hexaly and IB-LOP-Perm perform the experiments similarly. In case 5-7, Hexaly is faster than IB-LOP, which is faster than IB-LOP-Perm. Partly, this is due to the fact that IB-LOP-Perm starts by calculating the best order for client f, through a LOP evaluation in the base table of criteria, which is an additional step not applied in IB-LOP nor Hexaly. So, IB-LOP-Perm is less suitable to solve very small cases where calculating the standard LOP of the base table of criteria can take as much as the IB-LOP algorithm to solve the complete problem. In cases 5-11, 5-14, 5-18 and 5-21, IB-LOP-Perm already consumes less time than the IB-LOP, representing how the increase in complexity favours the heuristic. Hexaly obtains the same or better metrics than IB-LOP in all 5 client cases but 5-21, where the difference is of 8%. Hexaly also obtains the same or better metrics in 5-11 and 5-14 when compared to IB-LOP-Perm. However, as the criteria complexity increases, IB-LOP-Perm outperforms Hexaly in cases 5-18 and 5-21 while taking similar time of simulation.

When analysing 6 clients, the effect of the $O(n!)$ complexity in the IB-LOP starts to emerge. Now, IB-LOP-Perm is the best performing algorithm for all cases, followed by Hexaly, with exact IB-LOP as the last one. When compared to the 5-client cases, IB-LOP takes $\sim \times 5$ more time to solve the 6-client ones, while the difference for Hexaly is just a factor $\sim \times 2$ and for IB-LOP-Perm is practically non-existent.

These observations are confirmed and further accentuated when reaching 7 clients. Hexaly configurations consistently take roughly half the time required by IB-LOP, while all IB-LOP-Perm configurations take less than 5 s. For case 7-5, for instance, IB-LOP-Perm takes 19% of the time that Hexaly requires, and 10% of the time that IB-LOP needs. Alternatively, for case 7-21, the time savings are of 95% and 97% when comparing IB-LOP-Perm to Hexaly and IB-LOP, respectively. In addition, some of these 7 client cases already manifest how

IB-LOP-Perm works around the iteration complexity. Up to now, IB-LOP-Perm exercises for 5 and 6 clients where consistently taking less than 2 s, as only one iteration was needed due to their simplicity. However, while cases 7-11, 7-14 and 7-21 take around 2 s, cases 7-7 and 7-18 take approximately double. This means that, for these two cases, $Z_{pred} \neq Z_{check}$ in the first iteration, which resulted in a second iteration. This second iteration lasted similarly as the first one, and was enough to achieve Z^* for both cases.

In the cases for 8 clients, which is the memory limit for the IB-LOP, the time required rapidly increased to between 6 and 14 min for the exact algorithm. Now, an increase in the number of criteria visibly affects the complexity of the problem and time required to simulate it: while the difference in time between case 8-21 and 8-18 is of only 3%, adding criteria from case 8-11 to 8-14 and 8-14 to 8-18 produces a time increase of 25%, in each jump. Hexaly exercises also increase in time with the number of criteria, and regularly take around 60% less time than that required by IB-LOP, slightly increasing this difference with respect to the 7 client cases. Now, the IB-LOP-Perm is by far the fastest and most optimised resolution algorithm: for the hardest instance, case 7-21, it takes 99% less time than those required by both Hexaly and IB-LOP. As the complexity increases, so does the number of required iterations inside IB-LOP-Perm: for case 8-18 and 8-21, two iterations are not enough, so a third one is required. In any case, the time per iteration remains reasonable, being all the total times until reaching optimality below 7 s.

Therefore, when compared to IB-LOP and benchmarked with Hexaly, IB-LOP-Perm has proven to be the fastest and better scaling algorithm for the medium instances, especially in the 7 and 8 client cases where the increase in complexity is more notorious time-wise and between the 7, 11, 14, 18 and 21 criteria cases. Consequently, this study on medium instances justifies the use of IB-LOP-Perm for the cases on large instances.

7.3. Results on large instances

The experiments unmanageable by the exact algorithm on 10, 20 and 30 clients are represented in Table 4. The first three columns reflect the instance characteristics (number of clients, criteria and budget), while the fourth one is the current position of client f, Z_{IB-LOP}. The column of Z^* present in other tables disappears, as it cannot be calculated with the IB-LOP exact model. After, the best Zs obtained and the time taken when employing the random search scheme, the cheapest improvement greedy strategy and our IB-LOP-Perm heuristic are represented.

For the random search cases, the solutions generated did not improve the Z_{IB-LOP} that the client already had. In fact, most random search solutions were infeasible, as they did not comply with the feasible solution space constrained by the problem formulation. Indeed, this proves how IB-LOP-Perm and the greedy strategy rather efficiently direct the search of solutions in the solution space. The greedy strategy has proven to be useful, as it increases at least one position with respect to the current Z_{IB-LOP} everywhere except in cases 10-14 and 20-21. Some experiments present moderate increases of 1-3 units, like most cases with 10 clients and cases 20-11, 30-14. In other experiments, like cases 30-18, 30-11, 20-7 and 20-14 the increase is much higher. As the greedy strategy does not rely on the IB-LOP as a basis, its computation times are limited to the iterations needed until the budget runs out and no further cheapest improvement can be made. As a result, the greedy approach has the lowest simulation times, which do not vary significantly with the number of criteria nor when the number of clients increases.

Table 4. Comparison of greedy, random search and heuristic outcomes for large instances.

Problem definition				Random Search		Greedy		IB-LOP-Perm	
Clients	Criteria	Budget	Z_{IB-LOP}	Time (s)	Best Z	Time (s)	Best Z	Time (s)	Best Z
	7	18	3	300	3	5.1	4	5.3	8
	11	28	1	300	1	3.5	2	13.0	4
10	14	35	7	300	7	4.2	7	3.8	9
	18	121	0	300	0	5.9	3	4.4	9
	21	24	2	300	2	3.5	4	9.8	8
	7	61	9	600	9	3.5	16	13.7	19
	11	115	0	600	0	3.6	1	117.2	9
20	14	115	1	600	1	4.2	7	114.1	17
	18	140	0	600	0	3.2	6	39.2	18
	21	34	8	600	8	2.5	8	128.4	13
	7	67	3	1200	3	5.1	14	213.8	23
	11	153	13	1200	13	4.1	18	939.4	27
30	14	88	7	1200	7	4.7	9	230.1	23
	18	132	4	1200	4	8.7	16	1037.7	26
	21	103	4	1200	4	5.4	8	558.2	19

Nonetheless, IB-LOP-Perm is the method that consistently achieves a highest guess at the best Z. In all cases, it outperforms the greedy strategy and obtains a highest best Z than the current Z_{IB-LOP} one. In cases 10-11, 10-14 and 20-7 this difference is narrower, of around 1-3 positions, which represent a relative difference of 50%, 29% and 17%, respectively. The difference enlargens to around 4-7 positions in cases 10-7, 10-18, 10-21 and 20-21; for case 10-21, IB-LOP-Perm achieves a 200% improvement with respect to the greedy. In the rest of cases, more than 9 positions are improved when using IB-LOP-Perm with respect to the greedy strategy. In cases 20-14 and 30-18 the difference is of 10 positions, representing an improvement when using IB-LOP-Perm of 142% and 63%, respectively, when compared to the greedy. Even when the greedy strategy cannot find improvements to the current Z_{IB-LOP}, as in cases 10-14 and 20-21, IB-LOP-Perm is successful in finding at least 2 position improvements. However, the superior performance of the IB-LOP-Perm entails a computational complexity cost that is especially visible as the number of clients evolve: while the most demanding 10 client exercise took less than 15 s, the most demanding 30 client exercise required around 17 min. Not only the time required per iteration is increased due to the growing order of the preference matrix, but also the necessary iterations to achieve $Z_{pred} = Z_{check}$. In any case, for these large instances of the problem, IB-LOP-Perm proves its usefulness behaving the best in providing good quality solutions with respect to the random search and the greedy strategy.

8. An application on hospital rankings

To provide a real-life application, a case study has been developed about ranking hospitals. All the necessary inputs are obtained from the Health Reputation Monitor of 2024, which classifies hospitals in Spain according to 34 specialties (Merco, 2024c). The by-specialty rankings were developed by Corporate Reputation Business Monitor (Merco), a corporate monitor reference that has been evaluating the reputation of companies in Latin America since its launching in 2000 (Merco, 2024a). The rankings are based on surveys to hospital managers, doctors, nurses, medical journalists and patients, among others. Each specialty ranking is formed by a list of 10–15 hospitals, given a score between 3000, for the last one, up to 10000, for the best one. From the 34 available specialty rankings, a subset of 21 of them was sought, to keep coherence with the rest of exercises in the paper. From these 21 selected specialty rankings, an intersection of the medical centres common to all of them was performed. These operations resulted in a final table of 8 hospitals and 21 criteria. In Appendix A, Table A1 contains the abbreviations of the specialty rankings employed, Table A2 shows the abbreviations of the studied medical centres, and Table A3 illustrates the scores for each one of the 8 hospitals and 21 criteria. In this main body, Table 5 represents the final table of 8 hospitals and 21 criteria, represented also visually through the radar plot in Figure 1, with one coloured dashed line per medical centre, one grey radius line per criteria and one grey concentric circle per ranking position.

In this radar plot, smaller icosihenagons are associated to better positions across the criteria in the rankings, while icosihenagons covering larger surfaces mean worse positions. In general, these icosihenagons are quite irregular: for instance, while CLB is first in four criteria, it is also seventh (penultimate) in two criteria. This denotes some notable differences among the criteria and thus justifies our rank aggregation exercise. At plain sight, one of the best performing hospital seems LPZ, being first in ten criteria, although also the last one in C_{12}. On the contrary, the least performing one seems NAV, which is the last one in eight criteria but first in, interestingly, same C_{12}. For this reason, NAV has been chosen as targeted client f. The table of costs for each improvement, in Table 6, has been calculated for NAV by considering the cost of an improvement as the difference in score between the positions involved in the improvement, as extracted from Table A3 in Appendix A.

The results of the experiments are listed in Table 7, where cases have been built for four levels of budget and five levels of criteria: $C_1 - C_7$, $C_1 - C_{11}$, $C_1 - C_{14}$, $C_1 - C_{18}$ and $C_1 - C_{21}$. While the radar plot of the positions in $C_1 - C_{21}$ is already represented in Figure 1, the rest of the radar plots are

Table 5. Positions of the intersection of hospitals.

Abv.	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}	C_{21}
NAV	4	6	7	8	8	8	8	7	6	7	8	1	8	8	7	5	4	7	7	8	5
CLB	5	2	4	6	1	5	2	2	1	1	3	3	4	4	2	6	3	2	2	7	1
GRM	2	8	8	2	3	2	5	4	3	6	5	4	7	5	5	4	2	1	4	4	4
PLF	6	4	6	4	6	7	6	6	8	5	7	5	6	6	6	8	5	8	3	6	6
VDH	3	3	2	3	5	3	1	8	5	3	4	2	3	1	4	1	8	5	6	2	7
DDO	7	5	1	5	4	4	7	3	7	2	2	6	2	3	3	2	6	4	1	3	3
LPZ	1	1	3	1	2	1	4	1	2	4	1	8	1	2	1	3	1	3	5	1	2
VDR	8	7	5	7	7	6	3	5	4	8	6	7	5	7	8	7	7	6	8	5	8

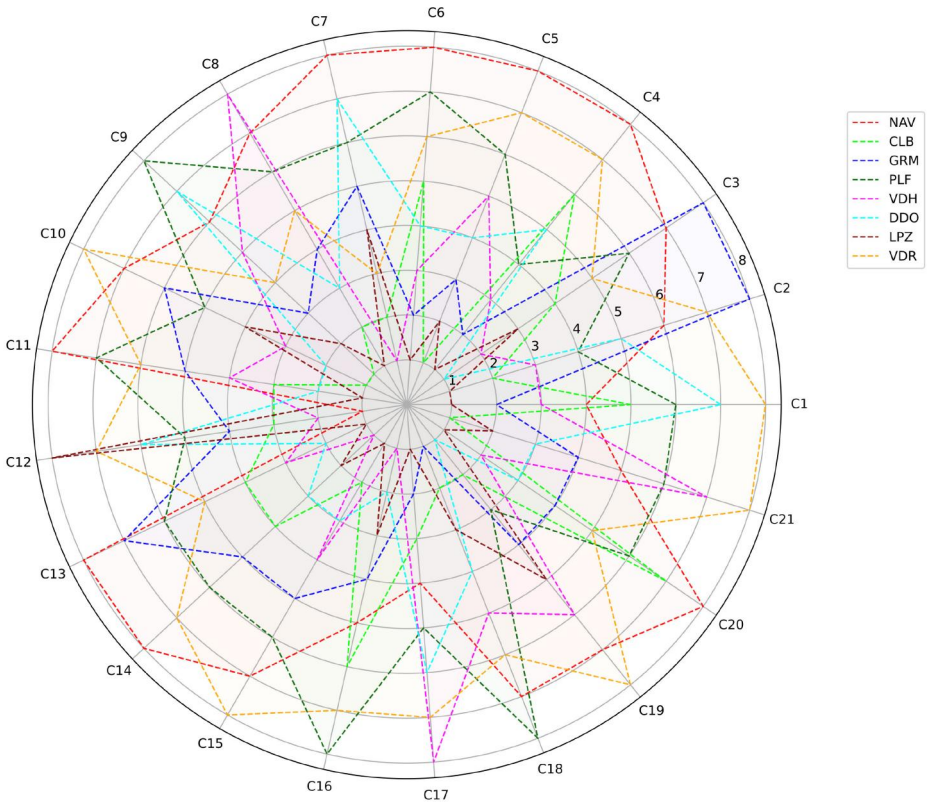


Figure 1. Radar plot of hospitals.

Table 6. Cost matrix of hospitals exercise when improving NAV.

Abv.	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}	C_{21}
2→1	2368	1548	1460	402	984	1555	771	2278	516	1904	2183	–	76	1882	1052	584	1298	3382	739	2678	210
3→2	1292	2211	412	2239	802	1868	80	238	1872	181	382	–	926	24	1592	2220	1632	498	281	636	1939
4→3	1298	796	1484	237	774	355	419	952	2216	163	1265	–	648	317	394	462	2072	1211	142	278	2259
5→4	–	163	1111	1301	746	217	527	1630	193	1136	41	–	1480	2491	1957	1369	–	541	610	162	358
6→5	–	757	1215	773	217	1826	419	274	614	223	1489	–	234	188	184	–	–	839	2273	908	–
7→6	–	–	18	590	591	574	897	674	–	491	1047	–	294	14	598	–	–	164	961	179	–
8→7	–	–	–	1332	2158	308	3316	–	–	–	188	–	3026	2068	–	–	–	–	–	1588	–

Table 7. Results of the hospitals exercise when improving NAV.

Problem definition				Time (s)		
Budget	Criteria	Z_{IB-LOP}	Z*	IB-LOP	Hexaly	IB-LOP-Perm
	7	0	3	676.7	330.4	5.3
	11	0	2	951.2	459.8	4.4
5000	14	0	2	1258.2	577.8	5.5
	18	0	2	1689.1	757.8	8.7
	21	0	2	1708.1	917.3	8.4
	7	0	4	796.4	365.2	4.1
	11	0	3	1114.6	489.4	5.1
10000	14	0	3	1398.2	679.1	5.9
	18	0	2	1710.6	795.5	11.4
	21	0	2	1894.5	903.2	8.8
	7	0	6	723.7	336.8	8.2
	11	0	5	1162.7	512.9	6.6
15000	14	0	4	1467.6	622.8	4.9
	18	0	4	1780.8	778.1	8.1
	21	0	3	1901.6	942.8	8.6
	7	0	7	703.8	336.3	8.5
	11	0	6	1216.3	502.7	3.6
20000	14	0	5	1475.5	650.5	11.3
	18	0	5	1890.2	800.9	10.5
	21	0	4	2165.7	895.7	14.2

found in Appendix B. Now, cases are referred to by their budget and number of criteria e.g., case 5000-18 has a budget of 5000 units and criteria $C_1 - C_{18}$. Across these five levels of criteria, $Z_{IB-LOP} = 0$, meaning that NAV is always the last one in the LOP of the unimproved table of criteria. Establishing budgets of 5000, 10000, 15000 and 20000 units provides a wide variety of Z^*, covering all possible cases except when $Z^* = 1$. Indeed, as the budget rises, higher Z^* are achievable: while budgets of 5000 can only achieve $Z^* = 2$ or $Z^* = 3$, a budget of 20000 can make NAV be the best hospital when considering only the first 7 criteria. A budget of 10000 units has the same effect as a budget of 5000 when considering 18 and 21 criteria, which is $Z^* = 2$, while it can improve one further position in the rest of cases: moving from $Z^* = 3$ to $Z^* = 4$ with 7 criteria, and from $Z^* = 2$ to $Z^* = 3$ when considering 11 and 14 criteria. When increasing the budget to 15000 units, all criteria configurations can improve at least one further position with respect to having 10000 units available to spend.

As in the exercises on 8 clients of the medium instances study, the three algorithms reach optimality, while IB-LOP-Perm performs faster than Hexaly,

than in turn performs faster than IB-LOP. Hexaly is capable of reducing 50%-60% the computational time required for the exact IB-LOP. Interestingly, increasing the budget and cost table to the order of magnitude of thousands of units has incremented the computational time, requiring IB-LOP 36 min to solve the most time demanding problem, 20000-21. Similarly, Hexaly needs 16 min to solve its most demanding problem, 15000-21. Though, this effect on the order of magnitude of the budget and cost table is not seen for the IB-LOP-Perm, which still reaches optimality below 15 s in all cases, requiring only 2 total iterations in the worst performing cases. This provides further evidence on the appropriateness and robustness of IB-LOP-Perm. Consequently, this application on hospital rankings not only provides a real-life justification of IB-LOP, but also strengthens the conclusions obtained on the results from the medium instances, as well as the usefulness of IB-LOP-Perm.

9. Conclusions and further research

In this work, we presented a novel extension of the Linear Ordering Problem (LOP), referred to as the Improvement Budgeted Linear Ordering Problem (IB-LOP). This model allows a designated client f to invest a limited budget to enhance their standing in one or more individual criteria, thereby potentially improving their position in the resulting aggregate ranking derived *via* the standard LOP formulation. The central objective of the IB-LOP is to determine the optimal set of improvements that client f should undertake within their budget to achieve the highest possible position in the global ranking.

We initially formulated the IB-LOP as a nonlinear bilevel mixed-integer programming model, which was subsequently transformed into a linear single-level model through standard reformulation techniques. During this process, we observed that the base IB-LOP model frequently admits multiple optimal solutions, some of which are preferable due to either lower cost or superior ranking performance in terms of the $\nu(\text{LOP})$ value. Consequently, we proposed two additional optimisation models to post-process the base IB-LOP solution: IB-LOP-MinC, which minimises the monetary cost required to achieve the same position, and IB-LOP-MaxL, which maximises the $\nu(\text{LOP})$ metric while preserving the improvement target. These models were evaluated on instances involving 2, 3, and 4 clients, across ranking scenarios with 7, 11, 14, 18, and 21 criteria. The resulting solutions varied across the three models, highlighting the relevance and practical utility of each formulation.

Due to the memory-intensive nature of the exact IB-LOP model, its application was limited to instances with up to 8 clients. To address this scalability issue, we developed a deterministic heuristic approach capable of efficiently handling larger instances. The heuristic was benchmarked against both the exact IB-LOP model and its implementation using Hexaly, for cases involving 5 to 8 clients. For 5 clients, all three methods exhibited comparable performance. For 6 and 7 clients, the heuristic consistently outperformed Hexaly, which remained faster than the exact method. This trend continued for 8 clients, with even more pronounced differences in computational time. Furthermore, when tested on larger-scale instances with 10, 20, and 30 clients, the heuristic significantly outperformed both a random search and a naive greedy approach, thereby validating its effectiveness. To demonstrate the practical implications of the IB-LOP, we also included a real-world case study on hospital rankings, where the objective was to strategically improve the position of a specific hospital by investing in selected medical specialty criteria.

Nonetheless, the study has certain limitations that warrant further investigation. On the side of the exact model, the primary constraint was RAM availability, which restricted the number of clients that could be realistically analysed. Enhancing computational resources or reformulating the model to reduce the number of variables or eliminate redundant solutions could mitigate this issue. Additional theoretical work may also uncover structural properties of the IB-LOP, especially concerning solution multiplicity and sensitivity of improvement costs, which could lead to more efficient formulations. With respect to the heuristic, future improvements may involve replacing the exact LOP solver with a heuristic one to further accelerate the IB-LOP. Moreover, dedicated heuristics for the IB-LOP-MaxL and IB-LOP-MinC variants could be developed, and alternative metaheuristic strategies such as genetic algorithms, simulated annealing, or tabu search could be explored in subsequent research.

Acknowledgments

This work was financially supported by the CIPROM/2024/34 grant, funded by the Conselleria de Educación, Cultura, Universidades y Empleo, Generalitat Valenciana. Moreover, the authors thank the grants PID2021-122344NB-I00 and PID2022-136383NB-I00, funded by MICIU/AEI/10.13039/501100011033 and by ERDF/EU. We also want to thank Professor Dolores Romero Morales for her support in the initial phase of the work.

Disclosure statement

No potential competing interest was reported by the authors.

ORCID

Iñigo Martín Melero http://orcid.org/0000-0001-7705-6051
Javier Alcaraz http://orcid.org/0000-0002-3526-508X

References

Alcaraz, J., García-Nové, E. M., Landete, M., & Monge, J. F. (2020). The linear ordering problem with clusters: A new partial ranking. *TOP*, *28*(3), 646–671. https://doi.org/10.1007/s11750-020-00552-3

Alcaraz, J., Landete, M., & Monge, J. F. (2022). Rank aggregation: Models and algorithms. In *The Palgrave handbook of operations research* (pp. 153–178). Springer.

Ali, A. B. (2024). Comprehensive learning TLBO with recursive precedence-based solution construction and multilevel local search for the linear ordering problem. *Expert Systems with Applications*, *238*, 122315.

Anderson, P. E., Chartier, T. P., Langville, A. N., & Pedings-Behling, K. E. (2022). Fairness and the set of optimal rankings for the linear ordering problem. *Optimization and Engineering*, *23*(3), 1289–1317. https://doi.org/10.1007/s11081-021-09650-y

Antuori, V., Hebrard, E., Huguet, M.-J., Essodaigui, S., & Nguyen, A. (2020). *Leveraging reinforcement learning, constraint programming and local search: A case study in car manufacturing* [Paper presentation]. In Principles and Practice of Constraint Programming: 26th International Conference, CP 2020, Louvain-la-Neuve, Belgium, September 7–11, 2020, Proceedings 26 (pp. 657–672). Springer.

Aparicio, J., Landete, M., & Monge, J. F. (2020). A linear ordering problem of sets. *Annals of Operations Research*, *288*(1), 45–64. https://doi.org/10.1007/s10479-019-03473-y

Aujac, H. (1960). La hiérarchie des industries dans un tableau des echanges industriels, rev,

Becker, O. (1967). *Das helmstädtersche reihenfolgeproblem—die effizienz verschiedener näherungsverfahren* [Paper presentation]. In Computer Uses in the Social Sciences, Berichteiner Working Conference.

Belloni, A., & Lucena, A. (2003). Lagrangian heuristics for the linear ordering problem. In *Metaheuristics: Computer decision-making* (pp. 37–63). Springer.

Benavides, X., Hernando, L., Ceberio, J., & Lozano, J. A. (2025). P-np instance decomposition based on the Fourier transform for solving the linear ordering problem. *Evolutionary Computation*, *33*(3), 395–423. https://doi.org/10.1162/evco_a_00368

Bertacco, L., Brunetta, L., & Fischetti, M. (2008). The linear ordering problem with cumulative costs. *European Journal of Operational Research*, *189*(3), 1345–1357. https://doi.org/10.1016/j.ejor.2006.03.071

Bornmann, L., Butz, A., & Wohlrabe, K. (2018). What are the top five journals in economics? A new meta-ranking. *Applied Economics*, *50*(6), 659–675. https://doi.org/10.1080/00036846.2017.1332753

Butler, T. D., Armstrong, C., Ellinger, A., & Franke, G. (2016). Employer trustworthiness, worker pride, and camaraderie as a source of competitive advantage: Evidence from great places to work. *Journal of Strategy and Management*, *9*(3), 322–343. https://doi.org/10.1108/JSMA-07-2015-0058

Bynum, M. L., Hackebeil, G. A., Hart, W. E., Laird, C. D., Nicholson, B. L., Siirola, J. D., Watson, J.-P., & Woodruff, D. L. (2021). *Pyomo–optimization modeling in python, volume 67* (3rd ed.). Springer Science & Business Media.

Cameron, T. R., Charmot, S., & Pulaj, J. (2021). On the linear ordering problem and the rankability of data. *arXiv Preprint*,

Campos, V., Glover, F., Laguna, M., & Martí, R. (2001). An experimental evaluation of a scatter search for the linear ordering problem. *Journal of Global Optimization*, *21*(4), 397–414. https://doi.org/10.1023/A:1012793906010

Campos, V., Martí, R. (2024). Optsicom project. https://grafo.etsii.urjc.es/optsicom/

Ceberio, J., Mendiburu, A., & Lozano, J. A. (2015). The linear ordering problem revisited. *European Journal of Operational Research*, *241*(3), 686–696. https://doi.org/10.1016/j.ejor.2014.09.041

Chanas, S., & Kobylański, P. (1996). A new heuristic algorithm solving the linear ordering problem. *Computational Optimization and Applications*, *6*(2), 191–205. https://doi.org/10.1007/BF00249646

Charon, I., & Hudry, O. (2010). An updated survey on the linear ordering problem for weighted or unweighted tournaments. *Annals of Operations Research*, *175*(1), 107–158. https://doi.org/10.1007/s10479-009-0648-7

Chen, W.-P., Chung, H., Lee, C., & Liao, W.-L. (2007). Corporate governance and equity liquidity: Analysis of S&P transparency and disclosure rankings. *Corporate Governance: An International Review*, *15*(4), 644–660. https://doi.org/10.1111/j.1467-8683.2007.00594.x

Chenery, H. B., & Watanabe, T. (1958). International comparisons of the structure of production. *Econometrica*, *26*(4), 487–521. https://doi.org/10.2307/1907514

Colmenar, J. M., Laguna, M., & Martín-Santamaría, R. (2024). Changeover minimization in the production of metal parts for car seats. *Computers & Industrial Engineering*, *198*, 110634. https://doi.org/10.1016/j.cie.2024.110634

Dakowska, D. (2013). Polish higher education and the global academic competition: University rankings in the reform debates. In *Global university rankings: Challenges for European higher education* (pp. 107–123). Palgrave Macmillan UK.

Diez-Cañamero, B., Bishara, T., Otegi-Olaso, J. R., Minguez, R., & Fernández, J. M. (2020). Measurement of corporate social responsibility: A review of corporate sustainability indexes, rankings and ratings. *Sustainability*, *12*(5), 2153. https://doi.org/10.3390/su12052153

Duan, W., Liang, D., & Zhang, Y. (2025). Towards improving decision quality for rank aggregation through consensus and utility dual-driven method. *Journal of the Operational Research Society*, *0*(0), 1–16. https://doi.org/10.1080/01605682.2025.2529533

Fernández Sánchez, J. L., Luna Sotorrío, L., & Baraibar Díez, E. (2012). Can corporate reputation protect companies' value? spanish evidence of the 2007 financial crash. *Corporate Reputation Review*, *15*(4), 228–239. https://doi.org/10.1057/crr.2012.13

Filbeck, G., Gorman, R., & Zhao, X. (2013). Are the best of the best better than the rest? the effect of multiple rankings on company value. *Review of Quantitative Finance and Accounting*, *41*(4), 695–722. https://doi.org/10.1007/s11156-012-0329-5

Fombrun, C. J. (1998). Indices of corporate reputation: An analysis of media rankings and social monitors' ratings. *Corporate Reputation Review*, *1*(4), 327–340. https://doi.org/10.1057/palgrave.crr.1540055

FT. (2025). Financial Times Business Education Regional Rankings. Retrieved January 8, 2025 from https://rankings.ft.com/business-education/regional-rankings

Gadd, E. (2020). University rankings need a rethink. *Nature*, *587*(7835), 523–523. https://doi.org/10.1038/d41586-020-03312-2

Garcia, C. G., Pérez-Brito, D., Campos, V., & Martí, R. (2006). Variable neighborhood search for the linear ordering problem. *Computers & Operations Research*, *33*(12), 3549–3565. https://doi.org/10.1016/j.cor.2005.03.032

Garcia, E., Ceberio, J., & Lozano, J. A. (2019). *Hybrid heuristics for the linear ordering problem* [Paper presentation]. 2019 IEEE Congress on Evolutionary Computation (CEC) (pp. 1431–1438). IEEE. https://doi.org/10.1109/CEC.2019.8790280

García-Nové, E. M., Alcaraz, J., Landete, M., Monge, J. F., & Puerto, J. (2017). Rank aggregation in cyclic sequences. *Optimization Letters*, *11*(4), 667–678. https://doi.org/10.1007/s11590-016-1047-z

Goglio, V. (2016). One size fits all? A different perspective on university rankings. *Journal of Higher Education Policy and Management*, *38*(2), 212–226. https://doi.org/10.1080/1360080X.2016.1150553

Gurobi Optimisation, LLC. (2024). Gurobi Optimizer Reference Manual. https://www.gurobi.com.

Hammarfelt, B., De Rijcke, S., & Wouters, P. (2017). From eminent men to excellent universities: University rankings as calculative devices. *Minerva*, *55*(4), 391–411. https://doi.org/10.1007/s11024-017-9329-x

Hart, W. E., Watson, J.-P., & Woodruff, D. L. (2011). Pyomo: Modeling and solving mathematical programs in python. *Mathematical Programming Computation*, *3*(3), 219–260. https://doi.org/10.1007/s12532-011-0026-8

Hautz, J., Hungerländer, P., Lechner, T., Maier, K., & Rescher, P. (2020). The weighted linear ordering problem. In *Operations Research Proceedings 2019: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Dresden, Germany, September 4-6, 2019* (pp. 223–229) Springer International Publishing.

Hexaly. (2025). Gurobi Vs, H. Retrieved May 4, 2025 from https://www.hexaly.com/gurobi

Hochbaum, D. S. (2006). Ranking sports teams and the inverse equal paths problem. In *Internet and Network Economics: Second International Workshop, WINE 2006, Patras, Greece, December 15-17, 2006. Proceedings*, volume 2, (pp. 307–318). Patras, Greece, Springer Berlin Heidelberg.

Hochbaum, D., Baumann, P., Goldschmidt, O., & Zhang, Y. (2025). A fast and effective breakpoints heuristic algorithm for the quadratic knapsack problem. *European Journal of Operational Research*, *323*(2), 425–440. https://doi.org/10.1016/j.ejor.2024.12.019

Horstschräer, J. (2012). University rankings in action? the importance of rankings and an excellence competition for university choice of high-ability students. *Economics of Education Review*, *31*(6), 1162–1176. https://doi.org/10.1016/j.econedurev.2012.07.018

Huang, G., & Lim, A. (2003). *Designing a hybrid genetic algorithm for the linear ordering problem* [Paper presentation].Genetic and Evolutionary Computation Conference (pp. 1053–1064). Springer.

Johnes, J. (2018). University rankings: What do they really show? *Scientometrics*, *115*(1), 585–606. https://doi.org/10.1007/s11192-018-2666-1

Kernighan, B. W., & Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, *49*(2), 291–307. https://doi.org/10.1002/j.1538-7305.1970.tb01770.x

Kirchler, M., Lindner, F., & Weitzel, U. (2018). Rankings and risk-taking in the finance industry. *The Journal of Finance*, *73*(5), 2271–2302. https://doi.org/10.1111/jofi.12701

Labbé, M., Landete, M., & Monge, J. F. (2023). Bilevel integer linear models for ranking items and sets. *Operations Research Perspectives*, *10*, 100271. https://doi.org/10.1016/j.orp.2023.100271

Laguna, M., Marti, R., & Campos, V. (1999). Intensification and diversification with elite tabu search solutions for the linear ordering problem. *Computers & Operations Research*, *26*(12), 1217–1230. https://doi.org/10.1016/S0305-0548(98)00104-X

Luca, M., & Smith, J. (2013). Salience in quality disclosure: Evidence from the us news college rankings. *Journal of Economics & Management Strategy*, *22*(1), 58–77.

Mahmood, A. (2020). Identifying the influence of various factor of apps on google play apps ratings. *Journal of Data, Information and Management*, *2*(1), 15–23. https://doi.org/10.1007/s42488-019-00015-w

Martí, R., & Reinelt, G. (2011). *The linear ordering problem: Exact and Heuristic methods in combinatorial optimization, volume 175*. Springer Science & Business Media.

Martí, R., & Reinelt, G. (2022). Exact and heuristic methods in combinatorial optimization. *Applied Mathematical Sciences*.

Martí, R., Reinelt, G., & Duarte, A. (2012). A benchmark library and a comparison of heuristic methods for the linear ordering problem. *Computational Optimization and Applications*, *51*(3), 1297–1317. https://doi.org/10.1007/s10589-010-9384-9

Martí, R., Sevaux, M., & Sörensen, K. (2024). 50 years of metaheuristics. *European Journal of Operational Research*, *321*(2), 345–362.

Martin-Melero, I. (2024). Instances for IB-LOP. https://github.com/inigo-martin-melero/1.Cases_UpgradedLOP.

Merco, M. (2024a). Retrieved March 4, 2025 from https://www.merco.info/es/que-es-merco

Merco, M. (2024b). Retrieved July 31, 2025 from https://www.merco.info/es/rankings-merco

Merco, M. (2024c). Monitor de Reputación Sanitaria – Servicios Hospitalarios. Retrieved March 4, 2025 from https://www.merco.info/es/monitor-reputacion-sanitaria-servicios-hospitalarios

Morales, J. A., Sánchez, S., Flores, J., Pineda, C., Gershenson, C., Cocho, G., Zizumbo, J., Rodríguez, R. F., & Iñiguez, G. (2016). Generic temporal features of performance rankings in sports and games. *EPJ Data Science*, *5*(1), 1–16. https://doi.org/10.1140/epjds/s13688-016-0096-y

Motegi, S., & Masuda, N. (2012). A network-based dynamical ranking system for competitive sports. *Scientific Reports*, *2*(1), 904. https://doi.org/10.1038/srep00904

NPR. (2010). Noma: A Look Back at Rene Redzepi's Revolution in Nordic Cuisine. NPR Online, October. Retrieved January 8, 2025 from https://www.npr.org/sections/pictureshow/2010/10/12/130517065/noma

Olcay, G. A., & Bulu, M. (2017). Is measuring the knowledge creation of universities possible?: A review of university rankings. *Technological Forecasting and Social Change*, *123*, 153–160. https://doi.org/10.1016/j.techfore.2016.03.029

Opferkuch, K., Caeiro, S., Salomone, R., & Ramos, T. B. (2022). Circular economy disclosure in corporate sustainability reports: The case of european companies in sustainability rankings. *Sustainable Production and*

Consumption, *32*, 436–456. https://doi.org/10.1016/j.spc.2022.05.003

Patara, S., & Dhalla, R. (2022). Sustainability reporting tools: Examining the merits of sustainability rankings. *Journal of Cleaner Production*, *366*, 132960. https://doi.org/10.1016/j.jclepro.2022.132960

QS. (2025a). Qs world university rankings. Retrieved July 31, 2025 from https://www.topuniversities.com/university-rankings

QS. (2025b, June). Qs world university rankings – big movers. Retrieved January 8, 2025 from https://www.qs.com/insights/articles/qs-world-university-rankings-big-movers/

Ratli, M., Urošević, D., El Cadi, A. A., Brimberg, J., Mladenović, N., & Todosijević, R. (2022). An efficient heuristic for a hub location routing problem. *Optimization Letters*, *16*, 281–300.

Sakuraba, C. S., Ronconi, D. P., Birgin, E. G., & Yagiura, M. (2015). Metaheuristics for large-scale instances of the linear ordering problem. *Expert Systems with Applications*, *42*(9), 4432–4442. https://doi.org/10.1016/j.eswa.2015.01.053

Schiavinotto, T., & Stützle, T. (2004). The linear ordering problem: Instances, search space analysis and algorithms. *Journal of Mathematical Modelling and Algorithms*, *3*(4), 367–402. https://doi.org/10.1023/B:JMMA.0000049426.06305.d8

Serenko, A., & Bontis, N. (2009). Global ranking of knowledge management and intellectual capital academic journals. *Journal of Knowledge Management*, *13*(1), 4–15. https://doi.org/10.1108/13673270910931125

Southern Cross University. (2024, June). Southern Cross University Tops Australia Rise in QS World University Rankings. Retrieved January 8, 2025 from https://www.scu.edu.au/news/2024/qs-most-improved-uni/

Statista. (2025). All rankings – Statista r. Retrieved July 7, 2025 from https://r.statista.com/en/all-rankings/

Szczecinski, L., & Sukheja, H. (2023). Rankability and linear ordering problem: Probabilistic insight and algorithms. *Computers & Operations Research*, *159*, 106347. https://doi.org/10.1016/j.cor.2023.106347

Taylor, P., & Braddock, R. (2007). International university ranking systems and the idea of university excellence. *Journal of Higher Education Policy and Management*, *29*(3), 245–260. https://doi.org/10.1080/13600800701457855

Thomas, H., & Peters, K. (2007). Business school rankings: Content and context. *Journal of Management Development*, *26*(1), 49–53.

Tischer, S., & Hildebrandt, L. (2014). Linking corporate reputation and shareholder value using the publication of reputation rankings. *Journal of Business Research*, *67*(5), 1007–1017. https://doi.org/10.1016/j.jbusres.2013.08.007

Tofallis, C. (2012). A different approach to university rankings. *Higher Education*, *63*(1), 1–18. https://doi.org/10.1007/s10734-011-9417-z

Trussel, J. M., & Patrick, P. A. (2018). Assessing and ranking the financial risk of municipal governments: The case of pennsylvania. *Journal of Applied Accounting Research*, *19*(1), 81–101. https://doi.org/10.1108/JAAR-05-2016-0051

Tsai, M.-F., & Wang, C.-J. (2013). In Advances in Information Retrieval: 35th European Conference on IR Research, ECIR 2013, Moscow, Russia, March 24-27, 2013. Proceedings, Risk ranking from financial reports, volume 35, pp. 804–807, Moscow, Russia, Springer Berlin Heidelberg.

University of Pretoria. (2023). UP improves its position by 20 places in 2024 QS World University Rankings on Sustainability to rank 156th globally. December. https://www.up.ac.za/department-of-institutional-advancement/news/post_3201428-up-improves-its-position-by-20-places-in-2024-qs-world-university-rankings-on-sustainability-to-rank-156th-globally, Retrieved January 8, 2025 from

Vaziri, B., Dabadghao, S., Yih, Y., & Morin, T. L. (2018). Properties of sports ranking methods. *Journal of the Operational Research Society*, *69*(5), 776–787. https://doi.org/10.1057/s41274-017-0266-8

Vieira, M. V. (2024). A linear ordering problem with weighted rank. *Journal of Combinatorial Optimization*, *47*(2), 13. https://doi.org/10.1007/s10878-024-01109-x

Appendix A.

Description of criteria and name of medical centres of hospital exercise.

Table A1. Description of criteria of hospital exercise.

Abv.	Criteria description
C_1	Allergology
C_2	Clinical analysis
C_3	Pathological anatomy
C_4	Anesthesiology and resuscitation
C_5	Digestive system
C_6	Cardiology
C_7	General surgery and digestive system
C_8	Medical-surgical dermatology and venereology
C_9	Endocrinology and nutrition
C_{10}	Hematology
C_{11}	Internal medicine
C_{12}	Nuclear medicine
C_{13}	Pneumology
C_{14}	Neurology
C_{15}	Obstetrics and gynaecology
C_{16}	Medical oncology
C_{17}	Otorhinolaryngology
C_{18}	Psychiatry
C_{19}	Radiodiagnosis
C_{20}	Traumatology and orthopaedic surgery
C_{21}	Urology

Table A2. Name of medical centres of hospital exercise.

Abv.	Hospital
NAV	Clínica Universidad de Navarra
CLB	Hospital Clínic Barcelona
GRM	Hospital General Universitario Gregorio Marañón
PLF	Hospital Universitari i Politècnic La Fe
VDH	Hospital Universitari Vall d'Hebron
DDO	Hospital Universitario 12 de Octubre
LPZ	Hospital Universitario La Paz
VDR	Hospital Universitario Virgen del Rocío

Table A3. Scores of the intersection of hospitals.

Abv.	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}	C_{21}
NAV	5042	4525	4300	3126	3492	3297	3571	3954	4589	5902	3405	10000	3316	3016	4223	5365	4998	3365	4994	3571	5234
CLB	4961	8452	6644	5048	9764	6005	9229	7722	10000	10000	7435	7372	8350	7777	8948	4848	7070	6618	9261	5159	10000
GRM	7632	4135	4262	9598	7978	8445	8203	6532	7612	6393	6129	7292	6342	5286	5005	6734	8702	10000	8838	6408	5592
PLF	4283	5445	4318	7122	6241	3605	7784	4628	3961	6616	3593	5290	6636	5098	4821	3720	4840	3162	8980	5338	4237
VDH	6340	6241	8540	7359	6458	6577	10000	3099	5203	7915	6170	9115	8998	10000	6962	10000	3235	4368	5955	7322	4123
DDO	4032	5282	10000	5821	7204	6222	6887	7484	4394	8096	7817	5188	9924	8094	7356	9416	4296	4909	10000	6686	7851
LPZ	10000	10000	8128	10000	8780	10000	8730	10000	9484	7752	10000	3748	10000	8118	10000	7196	10000	6120	8228	10000	9790
VDR	3797	4418	5533	4458	5650	4179	9149	4902	5396	5607	4640	4371	6870	5084	4058	4157	3355	3529	4614	6246	4033

Appendix B.

Radar plots on hospital exercises with 7, 11, 14 and 18 criteria

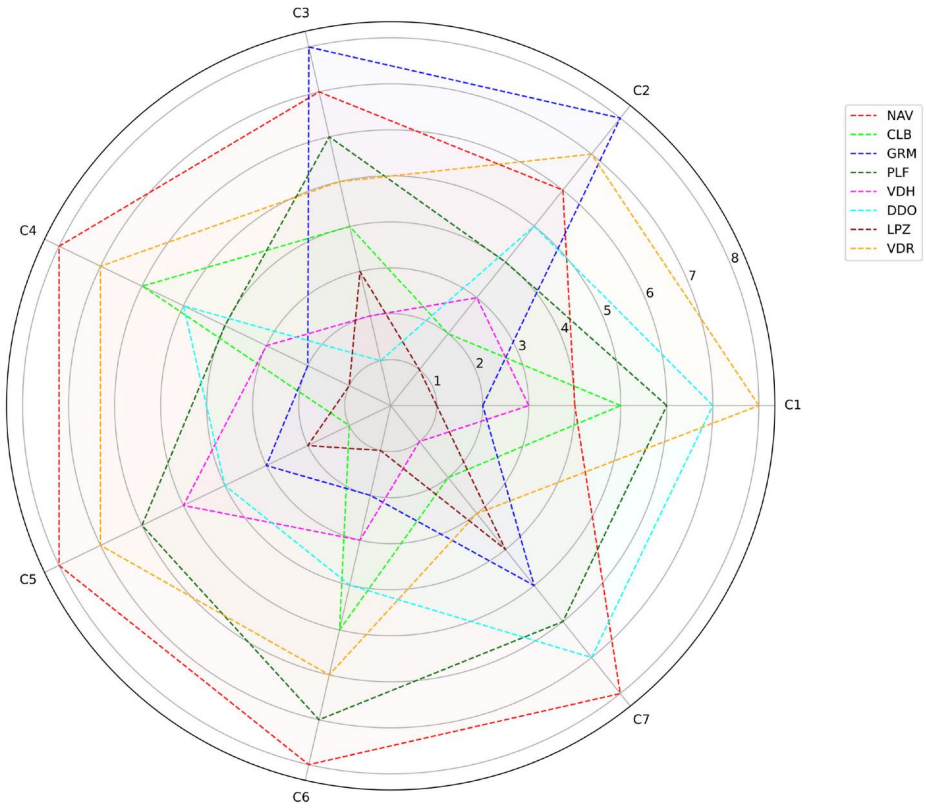


Figure B1. Radar plot of hospitals according to $C_1 - C_7$.

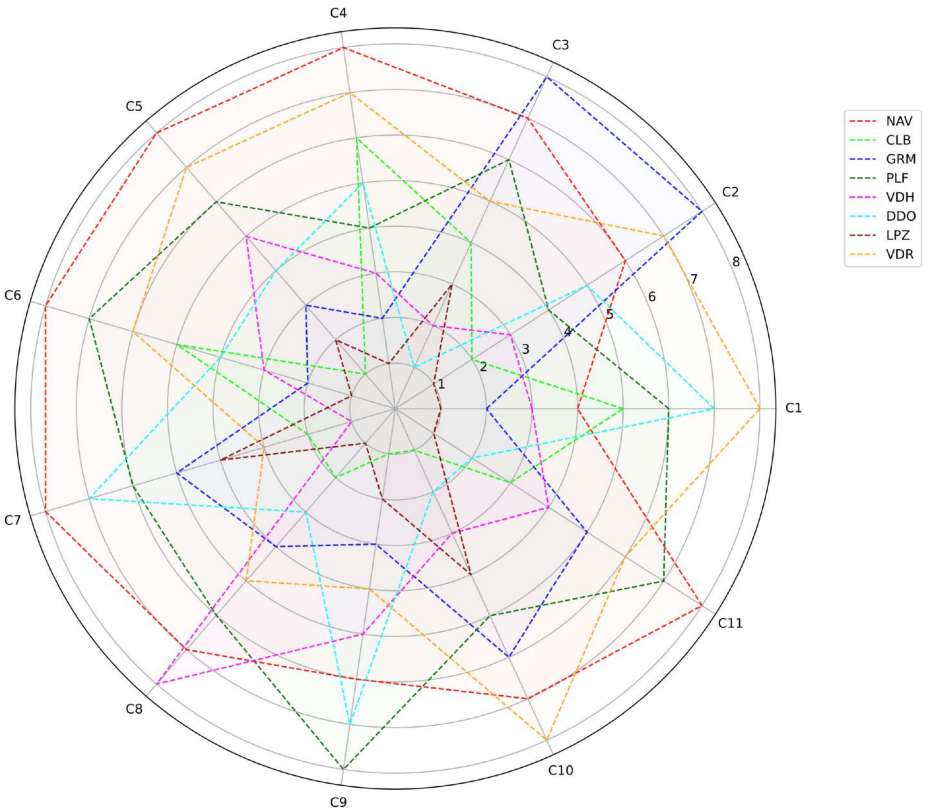


Figure B2. Radar plot of hospitals according to $C_1 - C_{11}$.

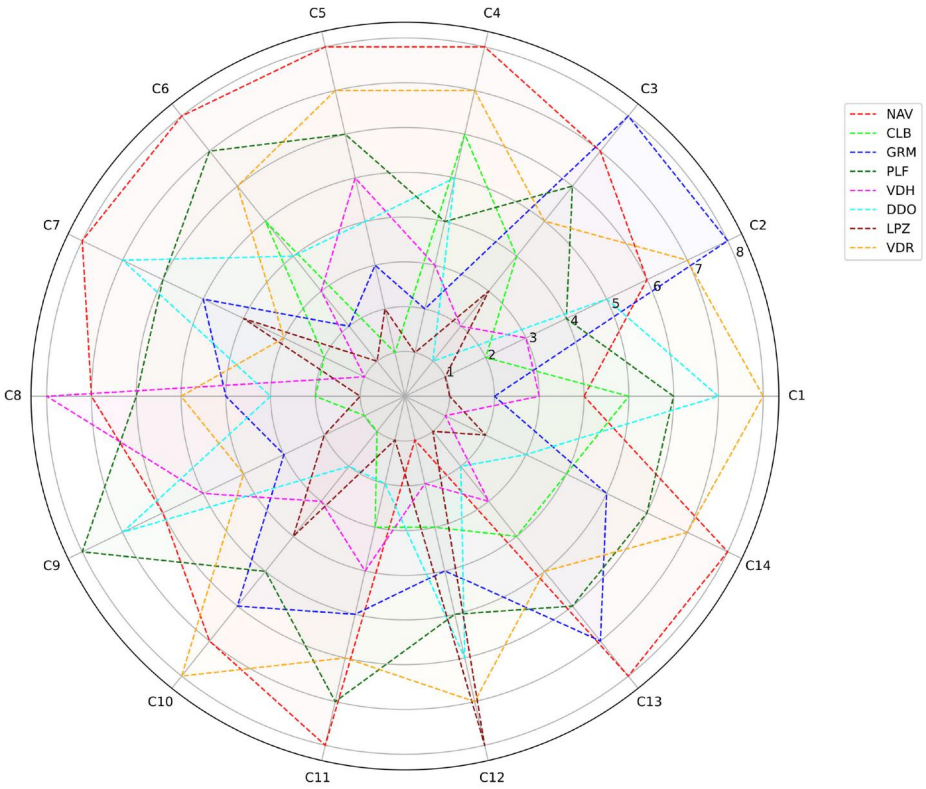


Figure B3. Radar plot of hospitals according to $C_1 - C_{14}$.

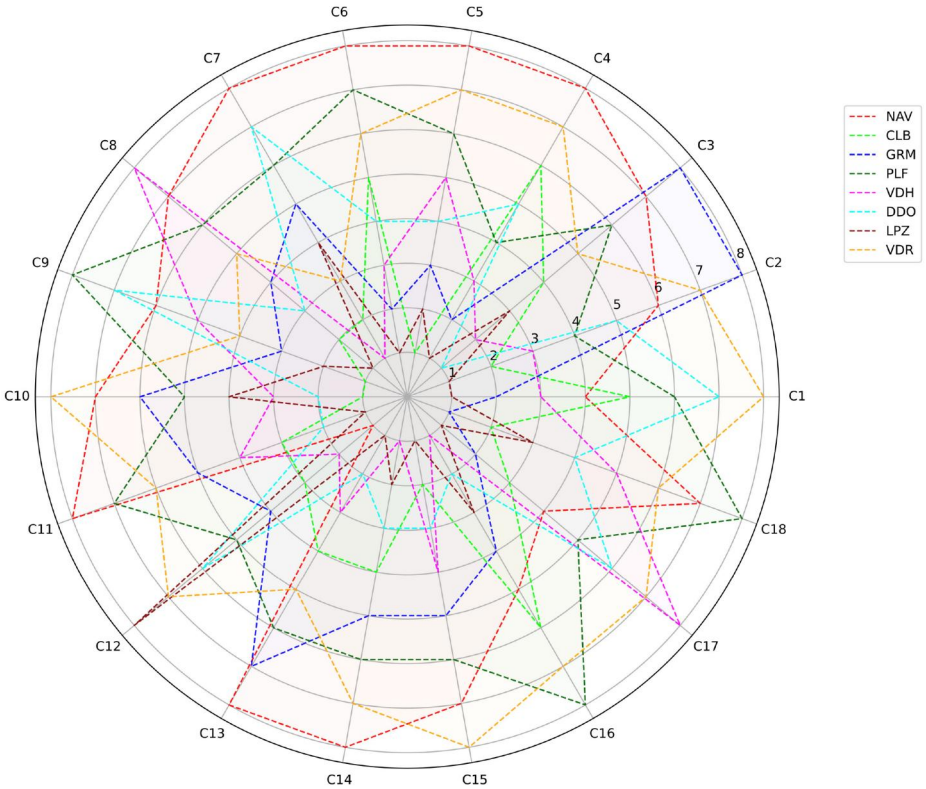


Figure B4. Radar plot of hospitals according to $C_1 - C_{18}$.