RAIRO-Oper. Res. 59 (2025) 803–833
https://doi.org/10.1051/ro/2025010

RAIRO Operations Research
www.rairo-ro.org

OPTIMIZING MULTILEVEL DECENTRALIZED PROBLEMS WITH A NOVEL APPLICATION OF GABASOV'S ADAPTIVE METHOD: HIGHLIGHTING APPLICATIONS IN COVID-19 VACCINE DISTRIBUTION

Mustapha Kaci*

Abstract. This paper presents a novel algorithm for solving multilevel monoobjective decentralized linear programming problems (ML-MO-DLPPs). Our approach represents a refined adaptation of Sinha and Sinha's linear programming method, enhanced by the development of an interval reduction map, which dynamically refines the decision variable intervals based on the influence of decisions made at preceding levels. Each stage of the algorithm's construction is thoroughly analyzed. The algorithm's effectiveness is demonstrated through a comprehensive numerical example, highlighting its practical applicability to resource management problems. Particular emphasis is placed on its application to vaccination planning in long-term care facilities during the COVID-19 pandemic, addressing the optimization of resource allocation with a strong focus on the equitable distribution of COVID-19 vaccines.

Mathematics Subject Classification. 90C05, 90C29, 90C51, 90C90.

Received January 27, 2024. Accepted February 6, 2025.

1. Introduction

Multilevel optimization, a robust framework for modeling decision making scenarios with dynamic interactions among multiple decision makers within a hierarchical structure, has found widespread applications across various sectors [1–3, 7, 16, 17, 23]. In multilevel mathematical programming problems, each level corresponds to a distinct decision maker, offering opportunities for cooperative optimization. Emphasizing collaboration and ensuring a minimum level of satisfaction for lower-level decision makers are essential for achieving overall organizational benefits. While bilevel programming has traditionally received significant attention, the landscape of programming problems extends well beyond this scope, encouraging researchers to explore innovative approaches to address the complexities of multilevel programming challenges [10, 11, 22, 25].

Multilevel decentralized programming problems (ML(D)PPs) have garnered increasing interest due to their applicability in diverse hierarchical decision making contexts. These problems represent a specific class of hierarchical optimization models characterized by multiple decision makers operating at different levels, each with distinct objectives and constraints. Unlike centralized problems, ML(D)PPs emphasize a decentralized

Keywords. Multilevel linear programming, Gabasov's adaptive method, simplex method, resource allocation, interval reduction map.

Laboratory of Mathematics, Djillali Liabes University of Sidi Bel-Abbés, P.O. Box 89, 22000 Sidi Bel-Abbés, Algeria.
*Corresponding author: mustapha.kaci@univ-sba.dz, kaci.mustapha.95@gmail.com

structure where decision making authority is distributed across various levels, reflecting real-world scenarios such as supply chain management, energy distribution, and policy planning. Decisions made by higher levels influence, and are influenced by, the actions of lower levels, creating a complex interplay between coordination and autonomy. Addressing these challenges requires methodologies that effectively account for the hierarchical and decentralized nature of the decision making process while ensuring consistency and alignment across all levels [16, 28].

1.1. Literature review

Over the past two decades, extensive exploration in the domain of multilevel programming/decision making has been documented through various survey papers [9, 12, 18, 26, 30]. These publications primarily focus on the early stages of research related to fundamental bilevel decision making, often within the confines of traditional solution approaches or specific application domains. The origins of multilevel decision making can be traced back to 1973 with the seminal paper by Bracken and McGill [10, 11]. Since the 1980s, a diverse body of related research has emerged, encompassing designations such as multilevel programming, multilevel optimization, and multilevel decision making.

Bilevel programming (BLP) represents a significant category in mathematical programming, offering a structured approach to model and solve optimization problems involving two levels of decision making. Numerous research efforts have been dedicated to exploring the theoretical and applied aspects of BLP, revealing its features, diverse applications, and associated solution methods.

In various contexts, BLP has been studied for its capabilities in formulating piecewise linear functions and its connections to other optimization problems, as extensively detailed in [9]. This paper not only reviews prior findings but also introduces new results, addressing the NP-hardness of BLP and clarifying confusing representations in the literature. Another comprehensive introduction to BLP is provided by Colson *et al.* [12], motivating this class through a simple application and detailing the general formulation of bilevel programs across linear, linear-quadratic, and nonlinear cases.

Special attention is given to applications in the energy networks domain, as outlined in [18]. New problems in this area, including natural gas cash-out, deregulated electricity market equilibrium, and biofuel design, are efficiently addressed as mixed-integer bilevel programs.

The survey in [26] encompasses both theoretical and applied results, exploring emerging themes such as migration processes, allocation problems, information protection, and cybersecurity. On the other hand, delves into decision making problems in decentralized organizations modeled as Stackelberg games, formulated as bilevel mathematical programming problems with two decision makers. The paper emphasizes the NP-hardness of solving such problems when decision makers lack motivation to cooperate. However, cooperative decision making is explored using interactive fuzzy programming, providing solutions aligned with decision makers' preferences. The paper extends these methods to bilevel linear programming problems in multiobjective environments and under uncertainty.

The bibliography compiled by Vicente and Calamai [30] stands as a dynamic and permanent contribution to bilevel and multilevel programming references. The authors invite suggestions for additions, corrections, and modifications. The overview of past and current research in bilevel and multilevel programming provided in the summary facilitates researchers' understanding of the references in this area. As a valuable resource, this bibliography supports and encourages further research in multilevel programming and decision making.

1.2. Recent advances

In recent years, the landscape of multilevel programming has witnessed the introduction of cutting-edge methodologies. Abo-Sinna's algorithm [1] targets the identification of α-Pareto optimal solutions, incorporating tolerance membership functions and exploring cooperative scenarios within the BLP problems with fuzzy parameters. Simultaneously, the work in [7] contributes two innovative algorithms employing fuzzy goal programming for addressing multiobjective linear programming problems in a multilevel framework, with a specific emphasis

on optimizing fuzzy goals for decision makers. Additionally, the paper in [6] proposes a fuzzy goal programming algorithm to navigate the intricacies of solving decentralized bilevel multiobjective programming problems. Sinha and Sinha [27] utilize the fuzzy mathematical programming approach to minimize objectives using linear membership functions. Building on this foundation, their subsequent work in [28] introduces a practical method for solving linear ML(D)PPs based on linear programming within a supervised search procedure.

Moreover, parallel efforts by Kaci and Radjef provides an intriguing perspective. In their paper [16], a new approach is developed to solve multilevel multiobjective linear programming problems (ML-MOLPPs). This approach, not considering the hierarchy of the problem, provides the set of all compromises. A simple criterion is established to test the feasibility of potential compromises, and a method based on Yu and Zeleny's approach is employed to generate the set of all compromises. In a subsequent work [17], the authors extend their exploration by introducing a novel algorithm for ML-MOLPPs. This algorithm, nested within the Gabasov's adaptive method of linear programming, begins by generating the set of all possible compromises (non-dominated solutions). Subsequently, it efficiently selects the best compromise among the potential settlements. Transforming the initial multilevel problem into an ML-MOLPP with bounded variables, the Gabasov's adaptive method is then applied, proving its efficiency compared to the simplex method.

In addition to these advancements, two notable alternative techniques for solving ML-MOLPPs have been proposed. The first technique, presented in [19], introduces an approach based on fuzzy goal programming (FGP), offering a simpler and computationally less intensive alternative to the algorithm proposed by Baky [7]. This technique transforms each objective function at each level into fuzzy goals, defining suitable membership functions and control vectors for each level's decision makers. The fussy goal programming approach is then employed to achieve the highest degree for each membership goal by minimizing the sum of negative deviational variables, ultimately providing a compromise optimal solution for ML-MOLPP. The second technique, detailed in [20], addresses fully neutrosophic multilevel multiobjective programming problems, employing a unique strategy to convert the problem into equivalent ML-MOLPP with crisp values. The proposed approach utilizes ranking functions and fuzzy goal programming, demonstrating simplicity and uniqueness in providing compromise optimal solutions for decision makers.

Furthermore, the work in [8] introduces a FGP algorithm for solving bilevel multiobjective programming problems with fuzzy demands. This algorithm addresses the experts' imprecise or fuzzy understandings of parameters in the problem formulation, assumed to be characterized as fuzzy numbers. The FGP algorithm satisfactorily achieves optimal solutions for all decision makers in fuzzy demand scenarios.

1.3. Impact of COVID-19 and associated optimization approaches

The unparalleled challenges posed by the COVID-19 pandemic have reshaped decision making scenarios across various domains. The dynamic and often unpredictable nature of the crisis, from the surge in healthcare demands to disruptions in education and supply chains, necessitates a new level of adaptability. Traditional decision making frameworks are strained under the weight of uncertainty and urgency inherent in the pandemic's complexities. It is in this landscape that optimization emerges as a linchpin, offering a systematic and data-driven approach to navigate the intricacies of resource allocation, logistics, and strategic planning. The urgency of the pandemic underscores the critical role of optimization in ensuring efficient, equitable, and timely decisions, making it an indispensable tool for addressing the unique challenges precipitated by COVID-19.

A primary major concern is effectively managing hospital resources amid a spike in infections, while minimizing disruption to regular healthcare services. A multiobjective, multi-period linear programming model was proposed to optimize the distribution of infected patients, the evacuation rate of non-infected patients, and the creation of new COVID-19 intensive care units, simultaneously minimizing the total distance traveled by infected patients, the maximum evacuation rate of non-infected patients, and the infectious risk to healthcare professionals [4].

The education sector has also been profoundly impacted, with the need to adopt new learning approaches during school closures. A study was conducted to understand the learning habits of students in India (Maharashtra) during the suspension of classes due to the pandemic. This research used a linear programming model to analyze the importance of teachers during online learning [24].

In the context of the blood supply chain, a study sought to optimally configure a multi-echelon blood supply chain network considering uncertainties related to demand, capacity, and blood disposal rates. This approach used a bi-objective mixed-integer linear programming model and interactive possibilistic programming to optimally address the problem considering the special conditions of the pandemic [5].

The pandemic has also had a significant impact on mental health, increasing the demand for psychological intervention services. A study combined statistical analyses and discrete optimization techniques to solve the problem of assigning patients to therapists for crisis intervention with a single tele-psychotherapy session. The integer programming model was validated with real-world data, and its results were applied in a volunteer program in Ecuador [21].

Finally, the equitable distribution of COVID-19 vaccines has become crucial for economic recovery, especially in developing countries. A mixed-integer linear programming model was proposed for equitable COVID-19 vaccine distribution, considering constraints such as specific refrigeration requirements for certain vaccines and the availability of storage capacity [29].

1.4. Proposed approach and contributions

This study introduces a novel algorithm tailored to solve ML-MO-DLPPs. Drawing inspiration from the linear programming method developed by Sinha and Sinha, our approach offers a modified framework enriched by the adaptive method pioneered by Gabasov *et al.* [13–15]. By incorporating the Gabasov's adaptive method, our algorithm aims to enhance the efficiency and resolution speed compared to conventional methods such as the primal simplex method.

Building upon Sinha and Sinha's foundational work [28], our enhanced algorithm introduces the interval reduction map to refine decision variable intervals influenced by preceding decision makers in the multilevel structure. Notably, our methodology leverages the Gabasov's adaptive method to replace the primal simplex method, striving for accelerated problem-solving. The Gabasov's adaptive method, known for its effectiveness in solving monoobjective linear programming problems with bounded variables, introduces a numerical constructiveness that aligns seamlessly with the multilevel linear programming challenges presented in this work.

This paper presents a comprehensive algorithmic development, validated through a numerical demonstration showcasing its efficacy and practical applicability, notably in resource management contexts, with a specific focus on optimizing resource allocation and ensuring the equitable distribution of COVID-19 vaccines, particularly in the domain of vaccination planning within long-term care facilities. The adaptability of the algorithm to real-world challenges is underscored. Additionally, our study contributes to optimization methods, offering insights into the trade-offs between traditional linear programming and our adapted approach. The comparative analysis in Table 6 sheds light on computational performance variations under different parameter configurations (α_1 and α_2), aiding in the selection of optimization approaches based on specific requirements and resource constraints. With a focus on addressing resource management challenges, especially amid the COVID-19 pandemic.

1.5. Paper organization

The organization of this paper is as follows: Section 2, titled formulation of ML-MO-DLPPs, provides a meticulous mathematical formulation of the multilevel decentralized linear programming problem, offering a detailed exploration of its structural components. Section 3 lays the foundational groundwork by presenting essential background information, thereby setting the stage for a comprehensive understanding of subsequent developments. Section 4 narrows its focus to the linear formulation specific to the pth level, unraveling the intricacies of our proposed approach. Section 5 delves into the details of an algorithmic solution for solving the ML-MO-DLPP, utilizing Gabasov's adaptive method. Shifting the spotlight to practical application. Section 6 showcases our approach in the context of vaccination planning within long-term care facilities, illustrated through a carefully chosen numerical example. Finally, Section 7, titled conclusion, synthesizes our contributions, summarizes key findings, and explores the broader implications of our work, providing a comprehensive conclusion to our exploration of the ML-MO-DLPPs and their real-world applications.

2. Formulation of a ML-MO-DLPPs

Let DM_p denote the decision maker at the pth level, where $p = 1 \dots P$ and $P \geq 2$. These decision makers (DMs) are tasked with controlling the decision variables $\overline{x}^p = x_{p,1}, \dots, x_{p,n_p} \in \mathbb{R}^{n_p}$, for $j = 1 \dots n_p$, and $n = n_1 + \dots + n_P$. The decision variable column vector x is defined as $x^T = (\overline{x}^1, \dots, \overline{x}^P)$.

Consider $c_p^{i,j}$ as constants for $i, p = 1, \dots, P$ and $j = 1, \dots, n_i$. The objective function $f_p : \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_P} \longrightarrow \mathbb{R}$, corresponding to the pth level of our ML-MO-DLPP, is expressed as:

$$f_p(x) = c_p^T x = \begin{pmatrix} c_p^{1,j} & c_p^{2,j} & \dots & c_p^{P,j} \end{pmatrix} x = \begin{pmatrix} c_p^{1,1} & \dots & c_p^{1,n_1} & c_p^{2,1} & \dots & c_p^{2,n_2} & \dots & c_p^{P,n_P} \end{pmatrix} x.$$

This can be further expanded as:

$$f_p(x) = c_p^{1,1} x_{1,1} + \dots + c_p^{1,n_1} x_{1,n_1} + c_p^{2,1} x_{2,1} + \dots + c_p^{2,n_2} x_{2,n_2} + \dots + \dots + c_p^{P,1} x_{P,1} + \dots + c_p^{P,n_P} x_{P,n_P}.$$

A P-level decentralized linear programming problem with a single objective function at each level can be formulated as follows:

$$\begin{aligned}
&\textbf{Level 1: } \underset{\overline{x}^1}{\mathrm{Max}}\ f_1(x) = \underset{\overline{x}^1}{\mathrm{Max}}\ f_1(\overline{x}^1, \overline{x}^2, \dots, \overline{x}^P) = \underset{\overline{x}^1}{\mathrm{Max}}\ c_1^T x, \\
&\qquad \text{such that } \overline{x}^2, \dots, \overline{x}^P \text{ solves:} \\
&\textbf{Level 2: } \underset{\overline{x}^2}{\mathrm{Max}}\ f_2(x) = \underset{\overline{x}^2}{\mathrm{Max}}\ f_2(\overline{x}^1, \overline{x}^2, \dots, \overline{x}^P) = \underset{\overline{x}^2}{\mathrm{Max}}\ c_2^T x, \\
&\quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
&\quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
&\quad \vdots \qquad \text{such that } \overline{x}^P \text{ solves:} \qquad \vdots \qquad\qquad \vdots \\
&\textbf{Level P: } \underset{\overline{x}^P}{\mathrm{Max}}\ f_P(x) = \underset{\overline{x}^P}{\mathrm{Max}}\ f_P(\overline{x}^1, \overline{x}^2, \dots, \overline{x}^P) = \underset{\overline{x}^P}{\mathrm{Max}}\ c_P^T x,
\end{aligned} \tag{1}$$

subject to:

$$x \in S := \{x \in \mathbb{R}^n \ : \ Ax \leq b \ , x \geq 0 \ , \ b \in \mathbb{R}^m\},$$

where $S \neq \emptyset$ is the convex feasible region, m is the number of constraints, A is an $m \times n$ matrix, and b is an m-dimensional vector.

3. Preliminaries

3.1. The Gabasov's adaptive method

Gabasov's adaptive method relies on the linear search technique to find an optimal distance within the feasible region S along a given direction. This approach, commonly employed in constrained optimization algorithms, involves several key steps. First, a direction $d \in \mathbb{R}^n$ is chosen to improve the objective function's value. Subsequently, the maximum distance that can be traveled along this direction without exiting the feasible region S is determined. A linear search is then conducted in the chosen direction d using the calculated maximum distance. The feasible solution x is updated by moving optimally along the direction d. These steps are repeated until a specified termination criterion is met.

Consider a linear programming problem with bounded variables, formulated in the canonical form as follows:

$$\begin{cases} \max\limits_{x} f(x) = c^T x \\ Ax = b \\ l \leq x \leq u, \end{cases} \tag{2}$$

where c, l, $u \in \mathbb{R}^n$, such that $\|l\| < \infty$ and $\|u\| < \infty$, and $b \in \mathbb{R}^m$. The matrix A has dimensions $m \times n$ with $m < n$.

Definition 3.1.

(1) A vector x satisfying the constraints of problem (2) is termed a *feasible solution* or simply a *plan*. The set of all feasible solutions is the *feasible region*, defined as:

$$S := \{x \in \mathbb{R}^n \ : \ Ax = b, \ l \leq x \leq u\}.$$

(2) Let J_B be a set of indices such that the cardinality of J_B is equal to m, then J_B is termed *support* if the matrix A_B is invertible.
(3) The pair $\{x, J_B\}$, consisting of a plan x and a support J_B, is a *supporting plan*. A supporting plan is *non-degenerate* if $l_j < x_j < u_j$, for all $j \in J_B$.
(4) A feasible solution x^0 is *optimal* if $f(x^0) = \max_{x \in S} f(x)$.
(5) x_ϵ is ϵ-*optimal* or *sub-optimal* if $f(x^0) - f(x_\epsilon) \leq \epsilon$, where x^0 is an optimal solution for problem (2), and $\epsilon \geq 0$.

The principle of Gabasov's adaptive method is as follows: start with an initial supporting plan $\{x, J_B\}$, which is a candidate solution satisfying the problem's constraints. Then, iterate to reach a new supporting plan $\{\bar{x}, \bar{J}_B\}$ while improving the objective function value, *i.e.*, by seeking to satisfy the inequality $f(\bar{x}) \geq f(x)$.

3.1.1. Suboptimality estimate

Let $\{x, J_B\}$ be a supporting plan for problem (2), and x^0 be any plan such that $x^0 = x + \Delta x$. The increase in the objective function is expressed as:

$$\Delta f(x) = f(x^0) - f(x) = c^T x^0 - c^T x = c^T \big(x^0 - x\big) = c^T \Delta x. \tag{3}$$

Furthermore, observe that: $A\Delta x = A\big(x^0 - x\big) = Ax^0 - Ax = b - b = 0$. This implies: $A_B \Delta x_B + A_N \Delta x_N = 0$. Consequently, Δx_B can be expressed in terms of Δx_N: $\Delta x_B = -A_B^{-1} A_N \Delta x_N$. By substituting Δx_B into equation (3), we obtain: $\Delta f(x) = c^T \Delta x = c_B^T \Delta x_B + c_N^T \Delta x_N = -c_B^T A_B^{-1} A_N \Delta x_N + c_N^T \Delta x_N$. This allows us to rewrite the increase in the objective function as:

$$\Delta f(x) = -\big(c_B^T A_B^{-1} A_N - c_N^T\big) \Delta x_N. \tag{4}$$

To simplify our analysis further, we denote the vectors K and E as follows: $K_B^T = c_B^T A_B^{-1}$, and $E^T = K^T A - c^T = (E_N^T, E_B^T)$, where

$$\begin{cases} E_N^T = c_B^T A_B^{-1} A_N - c_N^T, \\ E_B^T = c_B^T A_B^{-1} A_B - c_B^T = 0. \end{cases}$$

Finally, using these notations, formula (4) can be rewritten as follows:

$$\Delta f(x) = -E^T \Delta x_N = \sum_{j \in J_N} E_j \big(x_j - x_j^0\big). \tag{5}$$

Theorem 3.2 (Optimality criterion [14, 15])**.** *Let $\{x, J_B\}$ be a supporting plan of the problem (2). Then, the following relations are sufficient to ensure the optimality of the supporting plan $\{x, J_B\}$:*

$$\begin{cases} E_j \geq 0 & \text{if } \ x_j = l_j \\ E_j \leq 0 & \text{if } \ x_j = u_j \\ E_j = 0 & \text{if } \ l_j < x_j < u_j \end{cases}, \quad j \in J_N. \tag{6}$$

Moreover, these relations are also necessary when the supporting plan $\{x, J_B\}$ is non-degenerate.

Using Theorem 3.2, we have the following inequality: $x_j - u_j \leq x_j - x_j^0 \leq x_j - l_j$, $j \in J_N$. This allows us to establish the following upper bounds:

$$\begin{cases} E_j\big(x_j - x_j^0\big) \leq E_j(x_j - l_j), & \text{if } E_j > 0 \\ E_j\big(x_j - x_j^0\big) \leq E_j(x_j - u_j), & \text{if } E_j < 0. \end{cases}$$

Consequently, we derive the following upper bound, denoted as $\beta(x, J_B)$, termed as the suboptimality estimate:

$$\begin{aligned} f(x^0) - f(x) = \sum_{j \in J_N} E_j\big(x_j - x_j^0\big) &= \sum_{E_j > 0, j \in J_N} E_j\big(x_j - x_j^0\big) + \sum_{E_j < 0, j \in J_N} E_j\big(x_j - x_j^0\big) \\ &\leq \sum_{E_j > 0, j \in J_N} E_j(x_j - l_j) + \sum_{E_j < 0, j \in J_N} E_j(x_j - u_j) \\ &= \beta(x, J_B). \end{aligned}$$

Theorem 3.3 (Sufficient optimality condition [14,15]). *Let $\{x, J_B\}$ be a supporting plan of the problem (2), and ϵ be a strictly positive number. Then, the plan x is an ϵ-optimal solution if and only if $\beta(x, J_B) < \epsilon$.*

3.1.2. Comparison of Gabasov's adaptive method and primal simplex method

Gabosov's adaptive method offers notable advantages over primal simplex method, particularly in handling bounded variables – whether positive, negative, or mixed – without requiring transformation. This allows it to manipulate a reduced-size matrix when dealing with bounded constraints, optimizing the computational process by focusing on essential variables and constraints. As a result, the problem's dimensionality is reduced, leading to faster convergence with fewer iterations.

A key strength of this method lies in its ability to start from within the feasible region, characteristic of interior-point methods, which provides greater flexibility in choosing an initial point to initialize the method. This flexibility can further improve efficiency, especially in complex problems with multiple feasible solutions.

3.2. Multilevel model for efficient COVID-19 vaccine distribution in the United Kingdom

In Section 6, we address a multilevel allocation problem for the efficient management of resources during the COVID-19 pandemic in 2021 in the United Kingdom (UK). The problem is formulated using a three-level linear programming model, considering the roles of central, regional, and local management.

First level: central management, represented by the government of the UK, aims to maximize regional coordination and vaccine supply while minimizing overall costs. Possible decisions at this level include the quantity of vaccine doses allocated to the UK. Associated constraints relate to the national vaccine production capacity.

Second level: regional management, each region of the UK manages resource allocation based on specific needs and capacities. Possible decisions at this level include the number of hospital beds allocated to each region and the number of resources allocated to each hospital in the region. The objective is to maximize vaccine utilization by region. Constraints include regional hospital capacity, regional resource stock, bed needs by region, and resource needs by hospital in each region.

Third level: local management, each hospital within each region manages resource allocation to ensure adequate medical care for patients. Possible decisions at this level include the number of hospital beds allocated to each care unit in each hospital in the region. The objective is to maximize vaccine utilization by hospitals in the region. Constraints include hospital bed capacity for each care unit in each hospital, resource stock for each hospital in each region, and bed needs for each care unit.

This multilevel modeling allows for hierarchical resource management, with specific objectives at each management level. The associated figure illustrates the hierarchy and relationships between management levels,

showing how decisions and objectives at the central level impact the regional and local levels. This approach provides a structured framework for the optimal allocation of resources during the COVID-19 pandemic.

Algorithm 1: Gabasov's monocriteria adaptive method.

Input: Initial supporting plan $\{x, J_B\}$, parameter ϵ.

Output: Optimal supporting plan $\{x, J_B\}$ or ϵ-optimal supporting plan.

Step 1: Compute $K_B^T = c_B^T A_B^{-1}$, $E_j^T = K^T a_j - c_j^T$, $j \in J_N$.

Step 2: Compute $\beta(x, J_B)$.

Step 3: If $\beta(x, J_B) = 0$, then stop with the optimal supporting plan $\{x, J_B\}$.

Step 4: If $\beta(x, J_B) \leq \epsilon$, then stop with the ϵ-optimal supporting plan $\{x, J_B\}$.

Step 5: Determine the set of non-optimal indices:

$$J_{\text{NNO}} = \{j \in J_N : [E_j < 0, x_j < u_j] \text{ or } [E_j > 0, x_j > l_j]\}.$$

Step 6: Choose an index j_0 from J_{NNO} such that $|E_{j_0}| = \max_{j \in J_{\text{NNO}}} |E_{j_0}|$.

Step 7: Compute the admissible improvement direction d using the relations:

$$\begin{cases} d_{j_0} = -\text{sign}(E_{j_0}) \\ d_j = 0 \qquad\qquad\qquad \text{if } j \neq j_0,\ j \in J_N. \\ d_B = -A_B^{-1} A_{j_0} d_{j_0} \end{cases} \tag{7}$$

Step 8: Compute $\theta_{j_1} = \min\limits_{j \in J_B} \theta_j$, where θ_j is determined by the formula:

$$\theta_j = \begin{cases} \frac{u_j - x_j}{d_j}, & \text{if } d_j > 0 \\ \frac{l_j - x_j}{d_j}, & \text{if } d_j < 0 \\ \infty, & \text{if } d_j = 0. \end{cases} \tag{8}$$

. **Step 9:** Compute θ_{j_0} using the formula:

$$\theta_{j_0} = \begin{cases} x_{j_0} - l_{j_0} & \text{if } E_{j_0} > 0 \\ u_{j_0} - x_{j_0} & \text{if } E_{j_0} < 0. \end{cases} \tag{9}$$

Step 10: Compute $\theta^0 = \min\{\theta_{j_1}, \theta_{j_0}\}$.

Step 11: Compute $\overline{x} = x + \theta^0 d$.

Step 12: Compute $\beta(\overline{x}, J_B) = \beta(x, J_B) - \theta^0 |E_{j_0}|$.

Step 13: If $\beta(\overline{x}, J_B) = 0$, then stop with the optimal supporting plan $\{x, J_B\}$.

Step 14: If $\beta(\overline{x}, J_B) \leq \epsilon$, then stop with the $\epsilon-$optimal supporting plan $\{x, J_B\}$.

Step 15: If $\theta^0 = \theta_{j_0}$, then set $\overline{J}_B = J_B$.

Step 16: If $\theta^0 = \theta_{j_1}$, then set $\overline{J}_B = (J_B \backslash \{j_1\}) \cup \{j_0\}$.

Step 17: Set $x = \overline{x}$ and $J_B = \overline{J}_B$, and go to **Step 1**.

The application of the developed method on a specific case featuring linear constraints and objective functions will be meticulously detailed in Section 6.1. Subsequently, a practical numeric demonstration will be provided in Section 6.2, utilizing real-world data.

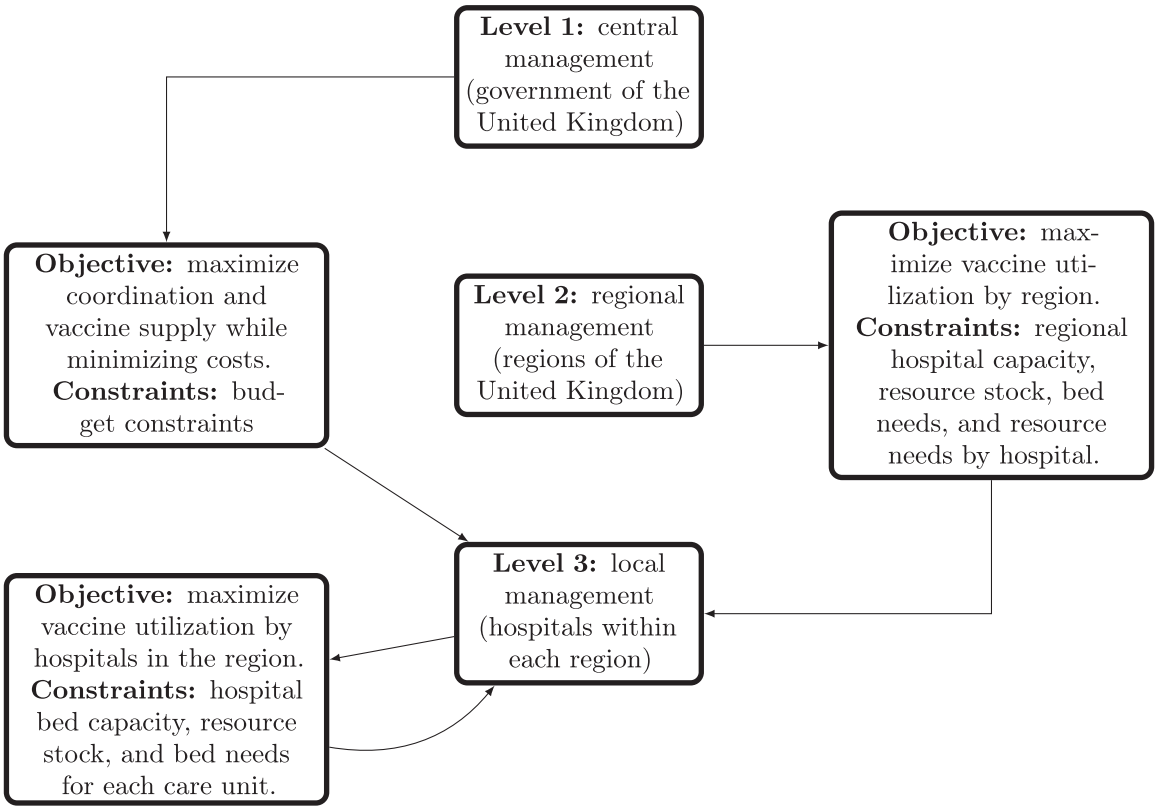


FIGURE 1. Detailed multilevel model for equitable COVID-19 vaccine distribution in the United Kingdom.

Remark 3.4. In the model depicted in Figure 1, the arrows indicate the relationships between the levels of management. Decisions and objectives at the central level (level 1) impact the regional (level 2) and local (level 3) levels. Resources and constraints propagate from the central level to the regional and local levels, where more detailed decisions are made to optimize vaccine distribution.

3.2.1. Why choose the UK for this study?

The choice of the UK for this study is motivated by several factors: its well-structured public health system (National Health Service, NHS), proven experience in managing major health crises, and the rapid implementation of a mass vaccination program in 2021. The UK features both centralized and decentralized governance, with regions having specific needs, which allows for the application and testing of a multilevel allocation model. Furthermore, the availability of detailed data on the pandemic and vaccine distribution makes it an ideal case study.

3.3. Resolution approach for decentralized multilevel problem (1)

Multilevel decentralized programming problems are marked by a hierarchical structure of DMs, where each decision level considers the decisions of the previous level to solve its own problem.

The following outline a practical approach to solving such problems:

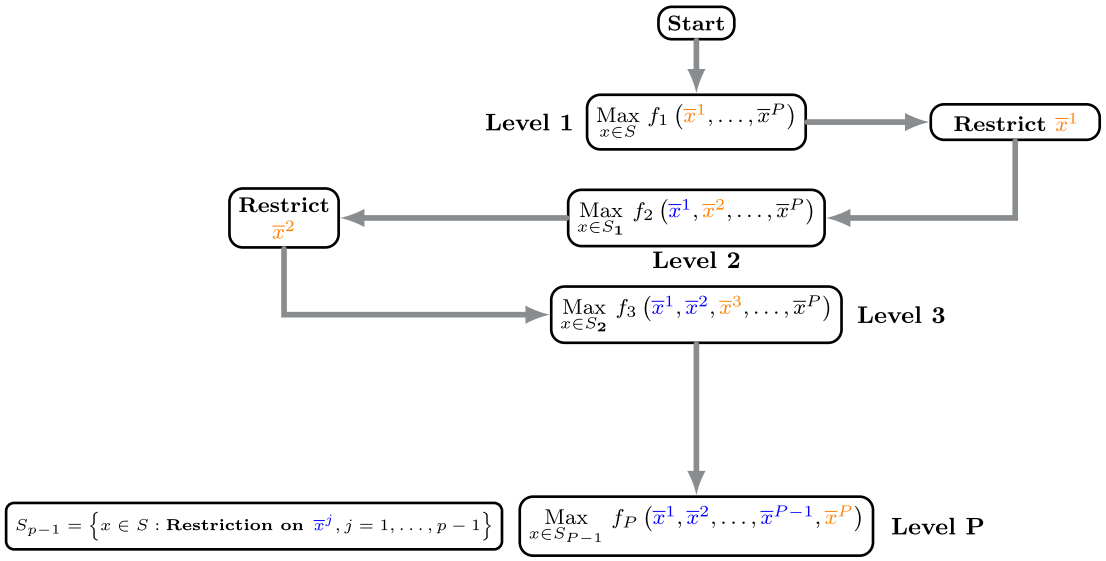


FIGURE 2. Resolution approach for a ML-MO-DLPP.

(1) **Definition of decision levels:** the first level (upper level) is the upper DM (DM_1), and the subsequent levels are subordinate DMs (DM_2, ... , DM_P). For this type of problem, the variables $\overline{x}^1, \overline{x}^2, \ldots, \overline{x}^P$ must adhere to global constraints that define the feasible region S of the multilevel problem.

(2) **Resolution of the upper DM' problem (upper level):** DM_1 solves its monoobjective problem considering only the initial constraints. Denoting f_1 as the objective function of the 1st level, the problem can be formulated as follows:

$$\underset{\overline{x}^1}{\text{Max}}\ f_1(x) = \underset{\overline{x}^1}{\text{Max}}\ f_1(\overline{x}^1, \overline{x}^2, \ldots, \overline{x}^P) = \underset{\overline{x}^1}{\text{Max}}\ c_1^T x, \quad x \in S, \tag{10}$$

where $\overline{x}^1$ are the DM_1's decision variables, and $\overline{x}^2, \ldots, \overline{x}^P$ represents the decision variables of subsequent levels.

(3) **Implementation of additional constraints in subordinate levels:** after attaining its optimal solution $\overset{c}{\overline{x}}^1$, the upper level utilizes this solution to introduce additional constraints in subordinate levels. These constraints narrow the search space down to a subset $S_1 \subset S$ of feasible solutions.

(4) **Resolution of subordinate DMs' problems:** subordinate DMs then solve their own monoobjective problems while considering the new constraints imposed by the upper levels. The DM_p solves their problem with S_1 as a search space for their decision variables. The problem for DM_p can be formulated as follows:

$$\underset{\overline{x}^p}{\text{Max}}\ f_p(x) = \underset{\overline{x}^p}{\text{Max}}\ f_p(\overline{x}^1, \ldots, \overline{x}^p, \ldots, \overline{x}^P) = \underset{\overline{x}^p}{\text{Max}}\ c_p^T x, \quad x \in S_1. \tag{11}$$

(5) **Iteration of the process:** the process continues by iterating steps 3 and 4 for each subordinate level until the last level is reached.

(6) **Attainment of compromise:** once all levels have solved their problems, progressively improving the solution from the upper level. The final compromise is reached at the last level, where all decisions are considered in a balanced manner, taking into account the constraints imposed by higher levels. Thus, the final compromise is the solution at the last level, reflecting the preferences of each decision level and resulting in an effective and balanced solution for the ML-MO-DLPP (Fig. 2).

3.4. Overview of Sinha and Sinha's method [28]

The paper [28] presents a linear programming approach to solving ML(D)PP. The method employs goal programming and a progressive reduction of decision intervals, providing a practical and straightforward solution by avoiding subjective elements often associated with fuzzy methods (such as tolerance values or membership functions).

Algorithm 2: Linear programming approach to solving ML(D)PPs.

Input: number of levels $P \geq 2$, m constraints, n decision variables.
Output: satisfactory solution after $P-1$ iterations.

1: **Identify** the multilevel problem with P levels, m constraints, and n decision variables.
2: **Compute** the optimal solutions of $f_p(x)$ and denote them as $\overset{\text{o}}{x}{}^{(1)}, \ldots, \overset{\text{o}}{x}{}^{(P)}$.
3: **Determine** the ideal range for each decision variable $x_{i,j}$, based on the optimal solutions from all levels:

$$l_{i,j}^{(0)} \leq x_{i,j} \leq u_{i,j}^{(0)}, \quad i = 1, 2, \ldots, P, \;\; j = 1, 2, \ldots, n_i.$$

4: **for** *each level* $p = 2$ *to* P **do**
(1) **Reduce** the ideal range of each decision variable controlled by the previous level's DM to narrow down feasible solutions. To do so, we follow the following 2 steps:
(a) **for** *each decision variable* $x_{i,j}$ *under the control of the* DM_{p-1} **do**
Decompose the ideal range into two subranges of the form: $\left[\overset{o}{x}{}_{i,j}^{(p-1)}, u_{i,j}^{(p-1)}\right]$, $\left[l_{i,j}^{(p-1)}, \overset{o}{x}{}_{i,j}^{(p-1)}\right]$, $\left[l_{i,j}^{(p-1)}, u_{i,j}^{(p-1)} - \alpha\right]$, or $\left[l_{i,j}^{(p-1)} + \alpha, u_{i,j}^{(p-1)}\right]$, where α is a positive number chosen by DM_{p-1}. Then, choose one of the subranges based on the sign of $x_{i,j}$ in $f_{p-1}(x)$ and the position of $\overset{o}{x}{}_{i,j}^{(p-1)}$.
end
(b) **Formulate** the chosen subranges as $2n_{p-1}$ constraints for pth level.
(2) **Solve** the single-objective linear problem of the pth level using all updated constraints to obtain a satisfactory solution $\overset{\text{c}}{x}{}^{p}$, then put $\overset{\text{o}}{x}{}^{(p)} = \overset{\text{c}}{x}{}^{p}$.
end

Return a satisfactory solution $\overset{\text{c}}{x}{}^{P}$ for the entire multilevel linear programming problem.

The linear programming approach in Algorithm 2 simplifies the search for a compromise solution in a hierarchical multilevel system by progressively reducing the range of each decision variable to satisfy individual and collective goals throughout the decision making hierarchy.

3.4.1. Description of each step

- **Problem identification**: formulate the multilevel programming problem by identifying each decision making level, their objective functions, decision variables, and constraints.
- **Individual optimization**: optimize the objective function of each level independently, resulting in an ideal solution for each decision maker.
- **Ideal ranges for variables**: define an ideal interval for each decision variable based on the minimum and maximum feasible values across all levels' optimal solutions.
- **Sequential range reduction**: the DM at the highest level begins by narrowing the interval of controlled variables, constraining the search space for lower levels.
- **Application of additional constraints**: apply the newly defined ranges as additional constraints for lower levels, progressively shrinking the decision space.

– **Iterative convergence**: continue this process to the lowest level, with each level applying constraints from higher levels and optimizing its objective function to reach a satisfactory solution.

3.4.2. Benefits of the approach

– **Practicality**: well-suited for real-world applications and avoids subjective parameters typically required in fuzzy programming.
– **Convergence**: achieves a single satisfactory solution while maintaining the hierarchical decision making structure.

Remark 3.5 (What is the nature of the solution?)**.** At each level p, the DM_p optimizes a single objective function, resulting in a truly optimal solution that achieves the maximum. Consequently, at each iteration, an optimal solution $\overset{\mathrm{c}\,p}{x}$ is determined in the classical sense, meaning:

$$f\left(\overset{\mathrm{c}\,p}{x}\right) \geq f(x), \quad \text{for all } x \in S_{p-1},$$

where

$$S_{p-1} = \left\{x \in S : \text{restrictions applied to } \overline{x}^j,\ j = 1, \ldots, p-1\right\}.$$

It is important to note that this approach does not rely on the concept of Pareto non-dominance. Instead, the solution is evaluated strictly based on classical optimization criteria at each level.

Thus, the compromise solution $\overset{\mathrm{c}\,P}{x}$ of the ML-MO-DLPP is an optimal solution for the Pth level linear single-objective programming problem.

3.5. Rationale for selecting the primal simplex method

The primal simplex method was chosen as a benchmark for comparing Gabasov's adaptive method due to its established reputation in linear optimization. Renowned for its robustness and efficiency, it remains a widely used standard for solving problems in canonical form, making it an ideal candidate for this study.

As a reference, the primal simplex highlights the distinctive strengths of Gabasov's method, particularly its ability to handle bounded variables directly without transformations and its constructive approach that reduces matrix dimensions and accelerates convergence.

This comparison aims to underscore the efficiency of Gabasov's adaptive method and its advantages in contexts where traditional methods, such as the primal simplex, might face limitations, particularly in hierarchical and multilevel optimization problems.

4. Linear formulation of the pth level of ML-MO-DLPP

In this section, we adapt the linear programming method proposed by Sinha and Sinha [28], as detailed in Algorithm 2, to reformulate the pth level of problem (1) into a single objective linear programming problem with bounded variables. This reformulation provides a foundation for the subsequent implementation of Gabasov's adaptive method (Algorithm 1).

Notations 4.1.

(1) $\overset{o(p)}{x}$ is the optimal solution of DM_p, and $\overset{o(p)}{x}_{i,j}$ is its (i,j)th component.
(2) $\mathrm{LW}_{p-1,j}$ and $\mathrm{UP}_{p-1,j}$ are the lower and upper intervals, respectively, of the $(p-1,j)$th component of the decision variable vector.

4.1. Construction of the interval reduction map

The application ξ_{p-1} that we will develop provides the reduced interval of decision variables controlled by DM_{p-1}, following the steps 3–9 precisely as proposed by Sinha and Sinha.

Notations 4.2.

(1) The superscript in the upper right ^{p} indicates that we are in the pth level, and the subscript in the lower right $_{i,j}$ designates the (i,j)th component.
(2) $l^{(p-1)}$ and $u^{(p-1)}$ represent the lower and upper bounds, respectively, of the decision variables at the pth level (these are the new bounds of the decision variables after reducing their intervals by DM_{p-1}). Notations $l_{i,j}^{(p-1)}$ and $u_{i,j}^{(p-1)}$ refer to their (i,j)th components.

The index $i = p-1$ signifies that we are addressing decision variables under the control of the DM_{p-1}.

Consider the multilevel linear programming problem (1), where $P \geq 2$. It includes m linear constraints, n decision variables, and a single objective function at each level. For $p = 1, \ldots, P$, let $\overset{o}{x}{}^{(p)}$ be the optimal solution of the following single-objective linear programming problem:

$$\begin{cases} \max\limits_{x} \; f_p(x) = c_p^T x \\ Ax \leq b \\ x \geq 0. \end{cases} \tag{12}$$

Next, we find the bounds for each of the n decision variables as follows:

$$l_{i,j}^{(0)} = \min_{q\in\{1,\ldots,P\}} \left\{ \overset{o}{x}{}_{i,j}^{(q)} \right\}, \quad u_{i,j}^{(0)} = \max_{q\in\{1,\ldots,P\}} \left\{ \overset{o}{x}{}_{i,j}^{(q)} \right\}, \qquad i = 1,\ldots,P, \quad j = 1,\ldots,n_i. \tag{13}$$

Then, set $l_{i,j}^{(1)} \leftarrow l_{i,j}^{(0)}$ and $u_{i,j}^{(1)} \leftarrow u_{i,j}^{(0)}$. Following this, consider the following $2n$ ideal intervals:

$$l_{i,j}^{(1)} \leq x_{i,j} \leq u_{i,j}^{(1)}, \qquad i = 1,\ldots,P, \quad j = 1,\ldots,n_i. \tag{14}$$

Guided by DM_p, each decision variable has the possibility of aligning with an ideal interval or reaching either of its lower or upper bounds. The following are the various scenarios:

$\overset{o}{x}{}_{p-1,j}^{(p-1)} = l_{p-1,j}^{(p-1)}$ — **3rd Case**

$l_{p-1,j}^{(p-1)} < \overset{o}{x}{}_{p-1,j}^{(p-1)} < u_{p-1,j}^{(p-1)}$ — **1st Case**

$\overset{o}{x}{}_{p-1,j}^{(p-1)} = u_{p-1,j}^{(p-1)}$ — **2nd Case**

Remark 4.3. Let $\delta_1, \delta_2, \delta_3 \in \mathbb{R}$ such that $\delta_1 < \delta_3 < \delta_2$, and define the applications

$$\begin{aligned} L: \; [\delta_1,\delta_2] &\longrightarrow [\delta_1,\delta_3] \\ x &\longrightarrow \frac{(\delta_3-\delta_1)x + \delta_1(\delta_2-\delta_3)}{\delta_2-\delta_1}, \end{aligned} \tag{15}$$

$$\begin{aligned} U: \; [\delta_1,\delta_2] &\longrightarrow [\delta_3,\delta_2] \\ x &\longrightarrow \frac{(\delta_2-\delta_3)x + \delta_2(\delta_3-\delta_1)}{\delta_2-\delta_1}. \end{aligned} \tag{16}$$

Then, L and U are bijective, continuous, and convex mappings that reduce the interval $[\delta_1,\delta_2]$ to $[\delta_1,\delta_3]$ (resp. $[\delta_3,\delta_2]$). This two maps are the key idea behind the construction of the interval reduction map ξ_{p-1}.

Case 1. In this case, the DM_{p-1} selects the reduced intervals of the decision variables by examining the sign before the decision variable $x_{p-1,j}$, $j=1,\dots,n_{p-1}$, in $f_{p-1}(x)$.

If the sign is negative, he choose the lower interval $\mathrm{LW}_{p-1,j}=\left[l_{p-1,j}^{(p-1)},\overset{o}{x}{}_{p-1,j}^{(p-1)}\right]$ using the application:

$$\begin{aligned} L_{p-1,j}:\quad \left[l_{p-1,j}^{(p-1)},u_{p-1,j}^{(p-1)}\right] &\longrightarrow \mathrm{LW}_{p-1,j}\\ x_{p-1,j} &\longrightarrow \frac{\left(\overset{o}{x}{}_{p-1,j}^{(p-1)}-l_{p-1,j}^{(p-1)}\right)x_{p-1,j}+l_{p-1,j}^{(p-1)}\left(u_{p-1,j}^{(p-1)}-\overset{o}{x}{}_{p-1,j}^{(p-1)}\right)}{u_{p-1,j}^{(p-1)}-l_{p-1,j}^{(p-1)}}. \end{aligned} \tag{17}$$

If the sign is positive, he choose the upper interval $\mathrm{UP}_{p-1,j}=\left[\overset{o}{x}{}_{p-1,j}^{(p-1)},u_{p-1,j}^{(p-1)}\right]$ using the application:

$$\begin{aligned} U_{p-1,j}:\quad \left[l_{p-1,j}^{(p-1)},u_{p-1,j}^{(p-1)}\right] &\longrightarrow \mathrm{UP}_{p-1,j}\\ x_{p-1,j} &\longrightarrow \frac{\left(u_{p-1,j}^{(p-1)}-\overset{o}{x}{}_{p-1,j}^{(p-1)}\right)x_{p-1,j}+u_{p-1,j}^{(p-1)}\left(\overset{o}{x}{}_{p-1,j}^{(p-1)}-l_{p-1,j}^{(p-1)}\right)}{u_{p-1,j}^{(p-1)}-l_{p-1,j}^{(p-1)}}. \end{aligned} \tag{18}$$

Note that applications $L_{p-1,j}$ and $U_{p-1,j}$ are defined similarly to those presented in (15) and (16), with the following parameters: $\delta_1=l_{p-1,j}^{(p-1)}$, $\delta_2=u_{p-1,j}^{(p-1)}$, $\delta_3=\overset{o}{x}{}_{p-1,j}^{(p-1)}$. These applications specifically affect the $(p-1,j)$th component.

Definition 4.4 (Sign function)**.** The "sign" function assigns the value $+1$ to positive numbers, -1 to negative numbers, and 0 to zero.

Proposition 4.5 (Definition)**.** *For a given decision variable $x_{p-1,j}$, where $x_{p-1,j}\in\left[l_{p-1,j}^{(p-1)},u_{p-1,j}^{(p-1)}\right]$, we define the following quantities:*

$$T_{p-1,j}^{-}=1-\mathrm{sign}\left(c_{p-1}^{p-1,j}\right)\quad\text{and}\quad T_{p-1,j}^{+}=1+\mathrm{sign}\left(c_{p-1}^{p-1,j}\right).$$

Then, we define the function $\psi_{p-1,j}(x_{p-1,j})$ as follows:

$$\psi_{p-1,j}(x_{p-1,j})=\frac{T_{p-1,j}^{-}L_{p-1,j}(x_{p-1,j})+T_{p-1,j}^{+}U_{p-1,j}(x_{p-1,j})}{2},$$

when $c_{p-1}^{p-1,j}\neq 0$, we get:

$$\psi_{p-1,j}\left(\left[l_{p-1,j}^{(p-1)},u_{p-1,j}^{(p-1)}\right]\right)=\begin{cases}\mathrm{LW}_{p-1,j} & \text{if } c_{p-1}^{p-1,j}<0,\\ \mathrm{UP}_{p-1,j} & \text{if } c_{p-1}^{p-1,j}>0.\end{cases}$$

Proof. Since $c_{p-1}^{p-1,j}\neq 0$, this means that ($c_{p-1}^{p-1,j}>0$ or $c_{p-1}^{p-1,j}<0$). Therefore, we have two cases to consider:

(1) When $c_{p-1}^{p-1,j}>0$, we get: $T_{p-1,j}^{+}=2$ and $T_{p-1,j}^{-}=0$. Then, the function $\psi_{p-1,j}$ becomes:

$$\psi_{p-1,j}\left(\left[l_{p-1,j}^{(p-1)},u_{p-1,j}^{(p-1)}\right]\right)=U_{p-1,j}\left(\left[l_{p-1,j}^{(p-1)},u_{p-1,j}^{(p-1)}\right]\right)=\mathrm{UP}_{p-1,j}.$$

(2) When $c_{p-1}^{p-1,j} < 0$, we get:

$$T_{p-1,j}^{+} = 0 \quad \text{and} \quad T_{p-1,j}^{-} = 2.$$

The function $\psi_{p-1,j}$ becomes:

$$\psi_{p-1,j}\left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right]\right) = L_{p-1,j}\left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right]\right) = \mathrm{LW}_{p-1,j}.$$

Proposition 4.7 explicitly addresses the situation when $c_{p-1}^{p-1,j} = 0$. □

Remark 4.6. We can observe that if $\mathrm{sign}(c_{p-1}^{p-1,j}) \neq 0$, then $\psi_{p-1,j}$ maps the interval $\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right]$ either to $\mathrm{LW}_{p-1,j}$ or to $\mathrm{UP}_{p-1,j}$. However, if $\mathrm{sign}(c_{p-1}^{p-1,j}) = 0$, it means that the component $x_{p-1,j}$ does not appear in the function f_{p-1} and thus does not influence the maximization of this function. In this case, we do not reduce the interval $\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right]$ because it would have no effect on the function's maximization. We simply apply the identity function to keep the interval unchanged.

It is important to note that these updates will only be applied if $c_p^{p-1,j} \neq 0$. Otherwise, the DM_p at the lower levels will not use them because the relevant variable does not appear in the function f_p.

All these considerations are taken into account by the application $\nu_{p-1,j}$, which handles the case where $c_p^{p-1,j} = 0$, as indicated in Proposition 4.7.

Proposition 4.7 (Definition)**.** *For a given decision variable $x_{p-1,j}$, where $x_{p-1,j} \in \left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right]$, we define the following quantities:*

$$S_{p-1}^{p-1,j} = \mathrm{sign}\left(c_{p-1}^{p-1,j}\right) \quad \text{and} \quad S_p^{p-1,j} = \mathrm{sign}\left(c_p^{p-1,j}\right).$$

Using the notation from Proposition 4.5, we then define:

$$\nu_{p-1,j}(x_{p-1,j}) = \left(S_p^{p-1,j} S_{p-1}^{p-1,j}\right)^2 \psi_{p-1,j}(x_{p-1,j}) + \left(1 - \left(S_p^{p-1,j} S_{p-1}^{p-1,j}\right)^2\right) x_{p-1,j}.$$

Thus, if both $c_{p-1}^{p-1,j} \neq 0$ and $c_p^{p-1,j} \neq 0$, we have:

$$\nu_{p-1,j}\left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right]\right) = \psi_{p-1,j}\left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right]\right).$$

Else, if $c_{p-1}^{p-1,j} = 0$ or $c_p^{p-1,j} = 0$, then we simply apply the identity application, that is:

$$\nu_{p-1,j} = Id_{\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right]}.$$

Proof.

(1) When both $c_{p-1}^{p-1,j} \neq 0$ and $c_p^{p-1,j} \neq 0$, we have:

$$\left(S_p^{p-1,j} S_{p-1}^{p-1,j}\right)^2 = 1, \quad 1 - \left(S_p^{p-1,j} S_{p-1}^{p-1,j}\right)^2 = 0$$

and

$$\nu_{p-1,j}\left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right]\right) = \psi_{p-1,j}\left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right]\right).$$

(2) Otherwise, if $c_{p-1}^{p-1,j} = 0$ or $c_p^{p-1,j} = 0$, then we simply apply the identity function, which means:

$$\left(S_p^{p-1,j} S_{p-1}^{p-1,j}\right)^2 = 0 \quad \text{and} \quad 1 - \left(S_p^{p-1,j} S_{p-1}^{p-1,j}\right)^2 = 1$$

and

$$\nu_{p-1,j}\left(\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right]\right) = \left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right] = Id_{\left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)}\right]}.$$

□

Case 2. If $\overset{o}{x}{}^{(p-1)}_{p-1,j} = u^{(p-1)}_{p-1,j}$, then the choice of interval reduction will be as follows:

$$\mathrm{LW}^{\alpha_{p-1,j}}_{p-1,j} = \left[l^{(p-1)}_{p-1,j}, u^{(p-1)}_{p-1,j} - \alpha_{p-1,j}\right],$$

using the application:

$$\begin{aligned} L^{\alpha_{p-1,j}}_{p-1,j} : \quad \left[l^{(p-1)}_{p-1,j}, u^{(p-1)}_{p-1,j}\right] &\longrightarrow \mathrm{LW}^{\alpha_{p-1,j}}_{p-1,j} \\ x_{p-1,j} &\longrightarrow \frac{\left(u^{(p-1)}_{p-1,j} - \left(l^{(p-1)}_{p-1,j} + \alpha_{p-1,j}\right)\right)x_{p-1,j} + \alpha_{p-1,j} l^{(p-1)}_{p-1,j}}{u^{(p-1)}_{p-1,j} - l^{(p-1)}_{p-1,j}}, \end{aligned} \tag{19}$$

where $\alpha_{p-1,j} \in \mathbb{R}$ is an arbitrary number chosen by DM_{p-1}, such that:

$$0 < \alpha_{p-1,j} < u^{(p-1)}_{p-1,j} - l^{(p-1)}_{p-1,j}. \tag{20}$$

Case 3. If $\overset{o}{x}{}^{(p-1)}_{p-1,j} = l^{(p-1)}_{p-1,j}$, then the choice of interval reduction will be as follows:

$$\mathrm{UP}^{\alpha_{p-1,j}}_{p-1,j} = \left[l^{(p-1)}_{p-1,j} + \alpha_{p-1,j}, u^{(p-1)}_{p-1,j}\right],$$

using the application:

$$\begin{aligned} U^{\alpha_{p-1,j}}_{p-1,j} : \quad \left[l^{(p-1)}_{p-1,j}, u^{(p-1)}_{p-1,j}\right] &\longrightarrow \mathrm{UP}^{\alpha_{p-1,j}}_{p-1,j} \\ x_{p-1,j} &\longrightarrow \frac{\left(u^{(p-1)}_{p-1,j} - \left(l^{(p-1)}_{p-1,j} + \alpha_{p-1,j}\right)\right)x_{p-1,j} + \alpha_{p-1,j} u^{(p-1)}_{p-1,j}}{u^{(p-1)}_{p-1,j} - l^{(p-1)}_{p-1,j}}. \end{aligned} \tag{21}$$

Proposition 4.8 (Definition). *For each component $\overset{o}{x}{}^{(p-1)}_{p-1,j}$, we define the following quantities:*

$$A_{p-1,j} = \mathrm{sign}\left(u^{(p-1)}_{p-1,j} - \overset{o}{x}{}^{(p-1)}_{p-1,j}\right) \text{ and } B_{p-1,j} = \mathrm{sign}\left(\overset{o}{x}{}^{(p-1)}_{p-1,j} - l^{(p-1)}_{p-1,j}\right).$$

Then, we define the function $\hat{\psi}_{p-1,j}$ as follows:

$$\hat{\psi}_{p-1,j}(x_{p-1,j}) = B_{p-1,j} L^{\alpha_{p-1,j}}_{p-1,j}(x_{p-1,j}) + A_{p-1,j} U^{\alpha_{p-1,j}}_{p-1,j}(x_{p-1,j}).$$

For any $x_{p-1,j} \in \left[l^{(p-1)}_{p-1,j}, u^{(p-1)}_{p-1,j}\right]$, we have:

- *If $A_{p-1,j} = 0$, then $\hat{\psi}_{p-1,j}(x_{p-1,j}) = L^{\alpha_{p-1,j}}_{p-1,j}(x_{p-1,j})$.*
- *If $B_{p-1,j} = 0$, then $\hat{\psi}_{p-1,j}(x_{p-1,j}) = U^{\alpha_{p-1,j}}_{p-1,j}(x_{p-1,j})$.*

Proof. This proposition is self-evident. □

Proposition 4.9 (Definition). *Let $x_{p-1,j} \in \left[l^{(p-1)}_{p-1,j}, u^{(p-1)}_{p-1,j}\right]$, using all the previous notations, we define:*

$$\hat{\nu}_{p-1,j}(x_{p-1,j}) = \left(S^{p-1,j}_{p} S^{p-1,j}_{p-1}\right)^2 \hat{\psi}_{p-1,j}(x_{p-1,j}) + \left(1 - \left(S^{p-1,j}_{p} S^{p-1,j}_{p-1}\right)^2\right) x_{p-1,j}.$$

Then,

$$\hat{\nu}_{p-1,j}\left(\left[l^{(p-1)}_{p-1,j}, u^{(p-1)}_{p-1,j}\right]\right) = \begin{cases} \hat{\psi}_{p-1,j}\left(\left[l^{(p-1)}_{p-1,j}, u^{(p-1)}_{p-1,j}\right]\right) & \text{if } c^{p-1,j}_{p-1} \neq 0 \text{ and } c^{p-1,j}_{p} \neq 0, \\ \hat{\nu}_{p-1,j} = Id_{\left[l^{(p-1)}_{p-1,j}, u^{(p-1)}_{p-1,j}\right]} & \text{else.} \end{cases}$$

Proof. The proof is similar to that of Proposition 4.7. □

Notations 4.10. Using all the previous notations, we denote the following:

1. $A^{+}_{p-1,j} = 1 + A_{p-1,j}$,
2. $A^{-}_{p-1,j} = 1 - A_{p-1,j}$,
3. $B^{-}_{p-1,j} = 1 - B_{p-1,j}$,
4. $B^{+}_{p-1,j} = 1 + B_{p-1,j}$,
5. $H^{1}_{p-1,j} = A^{+}_{p-1,j}A^{-}_{p-1,j} + B^{+}_{p-1,j}B^{-}_{p-1,j}$,
6. $H^{2}_{p-1,j} = -A_{p-1,j}B_{p-1,j}$.

Remark 4.11. In practice, in the 1st iteration at the 1st level, DM_1 solves its problem with respect to the initial constraints only, without any reduced interval of its variables. After obtaining its solution, it performs the reduction, which will be practically carried out in the second iteration by DM_2. Then, we define the interval reduction application at this level as follows: $\xi_0(x_{i,j}) = x_{i,j}$.

Definition 4.12 (Interval reduction map (IRM))**.** Let $i = 2, \ldots, P$, $j = 1, \ldots, n_i$, and $x_{i,j} \in \left[l^{(p-1)}_{i,j}, u^{(p-1)}_{i,j}\right]$. Then, we define:

$$\xi_{p-1}(x_{i,j}) = \begin{cases} H^{1}_{p-1,j}\hat{\psi}_{i,j}(x_{i,j}) + H^{2}_{p-1,j}\hat{\nu}_{i,j}(x_{i,j}) & \text{if } \ i = p-1, \\ x_{i,j} & \text{if } \ i \neq p-1. \end{cases} \tag{22}$$

Proposition 4.13. *The map ξ_{p-1} will adjust the decision variable intervals $x_{i,j}$ by taking into account all the modifications made previously.*

Proof. Let $i = 2, \ldots, P$, $j = 1, \ldots, n_i$ and $x_{i,j} \in \left[l^{(p-1)}_{i,j}, u^{(p-1)}_{i,j}\right]$, then

Proof of the first case: when $l^{(p-1)}_{p-1,j} < \overset{o}{x}{}^{(p-1)}_{p-1,j} < u^{(p-1)}_{p-1,j}$, we get:

$$\boxed{\begin{array}{l} A_{p-1,j} = 1, \\ B_{p-1,j} = -1 \end{array}} \Rightarrow \begin{cases} A^{+}_{p-1,j} = 1 + A_{p-1,j} = 2, \ \ A^{-}_{p-1,j} = 1 - A_{p-1,j} = 0, \\ B^{+}_{p-1,j} = 1 + B_{p-1,j} = 0, \ \ B^{-}_{p-1,j} = 1 - B_{p-1,j} = 2, \\ H^{1}_{p-1,j} = A^{+}_{p-1,j}A^{-}_{p-1,j} + B^{+}_{p-1,j}B^{-}_{p-1,j} = 2 \times 0 + 0 \times 2 = 0, \\ H^{2}_{p-1,j} = -A_{p-1,j}B_{p-1,j} \qquad\qquad\quad = -1 \times -1 \qquad = 1. \end{cases}$$

Then, we obtain:

$$\xi_{p-1}(x_{i,j}) = \begin{cases} 0 \times \hat{\psi}_{i,j}(x_{i,j}) + \hat{\nu}_{i,j}(x_{i,j}) = \hat{\nu}_{i,j}(x_{i,j}) & \text{if } \ i = p-1, \\ x_{i,j} & \text{if } \ i \neq p-1. \end{cases}$$

Proof of the second case: when $\overset{o}{x}{}^{(p-1)}_{p-1,j} = u^{(p-1)}_{p-1,j}$, we get:

$$\boxed{\begin{array}{l} A_{p-1,j} = 0, \\ B_{p-1,j} = 1 \end{array}} \Rightarrow \begin{cases} A^{+}_{p-1,j} = 1 + A_{p-1,j} = 1, \ \ A^{-}_{p-1,j} = 1 - A_{p-1,j} = 1, \\ B^{+}_{p-1,j} = 1 + B_{p-1,j} = 2, \ \ B^{-}_{p-1,j} = 1 - B_{p-1,j} = 0, \\ H^{1}_{p-1,j} = A^{+}_{p-1,j}A^{-}_{p-1,j} + B^{+}_{p-1,j}B^{-}_{p-1,j} = 1 \times 1 + 2 \times 0 = 1, \\ H^{2}_{p-1,j} = -A_{p-1,j}B_{p-1,j} \qquad\qquad\quad = -0 \times 1 \qquad = 0. \end{cases}$$

Then, we obtain:

$$\xi_{p-1}(x_{i,j}) = \begin{cases} \hat{\psi}_{i,j}(x_{i,j}) + 0 \times \hat{\nu}_{i,j}(x_{i,j}) = \hat{\psi}_{i,j}(x_{i,j}) & \text{if } \ i = p-1, \\ x_{i,j} & \text{if } \ i \neq p-1. \end{cases}$$

Proof of the third case: when $\overset{o}{x}{}^{(p-1)}_{p-1,j} = l^{(p-1)}_{p-1,j}$, we get:

$$\boxed{\begin{matrix} A_{p-1,j}=1, \\ B_{p-1,j}=0 \end{matrix}} \Rightarrow \begin{cases} A^{+}_{p-1,j} = 1 + A_{p-1,j} = 2, \;\; A^{-}_{p-1,j} = 1 - A_{p-1,j} = 0, \\ B^{+}_{p-1,j} = 1 + B_{p-1,j} = 1, \;\; B^{-}_{p-1,j} = 1 - B_{p-1,j} = 1, \\ H^{1}_{p-1,j} = A^{+}_{p-1,j}A^{-}_{p-1,j} + B^{+}_{p-1,j}B^{-}_{p-1,j} = 2 \times 0 + 1 \times 1 = 1, \\ H^{2}_{p-1,j} = -A_{p-1,j}B_{p-1,j} \qquad\qquad\quad = -1 \times 0 \qquad\;\; = 0. \end{cases}$$

Then, we obtain:

$$\xi_{p-1}(x_{i,j}) = \begin{cases} \hat{\psi}_{i,j}(x_{i,j}) + 0 \times \hat{\nu}_{i,j}(x_{i,j}) = \hat{\psi}_{i,j}(x_{i,j}) & \text{if } \; i = p-1, \\ x_{i,j} & \text{if } \; i \neq p-1. \end{cases}$$

□

Remark 4.14. The application ξ_{p-1} will reduce the intervals of the components $x_{p-1,j}$ for all $j = 1, \ldots, n_p$ only.

Notations 4.15. We denote by $l^{(p)}$ and $u^{(p)}$ the following vectors:

$$l^{(p)} := \xi_{p-1}\Big(l^{(p-1)}\Big) = \Big(\xi_{p-1}\Big(l^{(p-1)}_{i,j}\Big), i = 1, \ldots, P, j = 1, \ldots, n_i\Big), \tag{23}$$

$$u^{(p)} := \xi_{p-1}\Big(u^{(p-1)}\Big) = \Big(\xi_{p-1}\Big(u^{(p-1)}_{i,j}\Big), i = 1, \ldots, P, j = 1, \ldots, n_i\Big). \tag{24}$$

Definition 4.16. The linear formulation of the pth level problem is given as follows:

$$\begin{cases} \max\limits_{x} \; f_p(x) = c_p^T x \\ Ax \leq b \\ \xi_{p-1}\big(l^{(p-1)}\big) \leq x \leq \xi_{p-1}\big(u^{(p-1)}\big). \end{cases} \tag{25}$$

5. Algorithm to solve the ML-MO-DLPP (1) using the Gabasov's adaptive method Algorithm 1

The algorithm follows a systematic procedure. Initially, individual solutions for each objective function are computed independently, providing insight into the optimization landscape for each criterion. Following this, careful consideration is given to establishing bounds for all decision variables, a crucial step that sets the stage for subsequent computations. The compromise search phase is then initiated, involving the solution of $P-1$ standard linear programming problems with bounded variables using the adaptive method Algorithm 1.

Consider the following linear programming problem:

$$\begin{cases} \max\limits_{x} \; f_p(x) = c_p^T x \\ Ax \leq b \\ l^{(p)} \leq x \leq u^{(p)}. \end{cases} \tag{26}$$

Algorithm 3: Procedure for resolving the ML-MO-DLPP.

Step 1: set $p = 1$.
While $p \leq P$**, repeat the following steps:**

(1) explicit the model (12).
(2) obtain the solution $\overset{o(p)}{x}$ of the model (12) "optimal solution of the DM_p".
(3) set $p \leftarrow p + 1$.

End while
Step 2: obtain all bounds $l_{i,j}^{(1)}$ and $u_{i,j}^{(1)}$ using formula (13).
Step 3: set $p \leftarrow 2$.
While $p \leq P$**, repeat the following steps:**

(1) explicit the model (26).
(2) solve the model (26) using the Gabasov's monocriteria adaptive method Algorithm 1 to obtain the optimal solution $\overset{cp}{x}$.
(3) set $p \leftarrow p + 1$.

End while
Output: the satisfactory solution $\overset{cP}{x}$ for the multilevel problem.

6. Numerical example: Vaccination planning in long-term care facilities

Vaccination planning in long-term care facilities (LTCFs) presents a formidable challenge, demanding meticulous consideration of intricate variables such as facility capacity, demographics of the target population, and the specific requirements of vulnerable cohorts. This study aims to elucidate the complexities inherent in the UK's LTCF vaccination strategy for the year 2021. A mathematical model is deployed to simulate the distribution of COVID-19 vaccines across varied facilities, systematically incorporating constraints such as capacity limitations and the presence of vulnerable populations. The data employed in the model are derived from publicly available sources.

6.1. Model formulation

The analysis is regionally structured, with the UK segregated into four distinct administrative regions: England (Region 1), Scotland (Region 2), Wales (Region 3), and Northern Ireland (Region 4). Each region, denoted by $i = 1, 2, 3, 4$, is characterized by unique demographic and healthcare parameters. A detailed examination of hospitals within each region is undertaken, integrating information on vulnerable populations and care unit capacities, as presented in Table 1.

For clarity, in Region 1 (England), hospitals are labeled as 11 (Royal Free Hospital – RFH), 12 (University College Hospital – UCLH), 13 (University Hospitals of Leicester – UHL), 14 (Addenbrooke's Hospital – ADH), 15 (Guy's and St Thomas' NHS Foundation Trust – GSTT), 16 (Imperial College Healthcare NHS Trust – ICH), 17 (University Hospitals Birmingham NHS Trust – UHB), 18 (Manchester University NHS Foundation Trust – MFT), with $j_1 = 8$.

In Region 2 (Scotland), hospitals are denoted as 21 (Queen Elizabeth University Hospital – QEH), 22 (Ninewells Hospital and Medical School – 9W), 23 (Glasgow Royal Infirmary – GRI), 24 (Aberdeen Royal Infirmary – ARI), 25 (Royal Edinburgh Hospital – REH), with $j_2 = 5$.

Region 3 (Wales) is represented by hospital 31 (Ysbyty Gwynedd – YG), 32 (University Hospital of Wales – UHW), 33 (Morriston Hospital – MH), 34 (Royal Gwent Hospital – RGH), 35 (Wrexham Maelor Hospital – WMH), with $j_3 = 5$.

In Region 4 (Northern Ireland), features hospital 41 (Belfast City Hospital – BCH), 42 (Altnagelvin Area Hospital – AAH), 43 (Craigavon Area Hospital – CAH), 44 (South West Acute Hospital – SWAH), with $j_4 = 4$.

TABLE 1. Comprehensive real-world data for UK: population, COVID-19 cases, vaccination capacity, and healthcare infrastructure in 2021.

Region (i)	Population (P_i) (millions)	Cases (I_i) (millions)	Administration doses Capacity (C_i) (millions/year)	Equitable vaccine distribution (E_i) (%)
1-England	56.48	8.55	51.2	80
2-Scotland	5.45	0.45	4.95	70
3-Wales	3.16	0.15	2.87	60
4-Northern Ireland	1.91	0.1	1.78	50

Region (i)	Hospital (j)	Vulnerable population (V_{ij}) (millions)	Care hospital capacity (B_{ij}) beds (millions)
1-England	1-Royal Free Hospital (RFH)	0.025	0.030
	2-University College Hospital (UCLH)	0.030	0.035
	3-University Hospitals of Leicester (UHL)	0.020	0.025
	4-Addenbrooke's Hospital (ADH)	0.028	0.032
	5-Guy's and St Thomas' NHS Foundation Trust (GSTT)	0.032	0.034
	6-Imperial College Healthcare NHS Trust (ICH)	0.027	0.030
	7-University Hospitals Birmingham NHS Trust (UHB)	0.035	0.038
	8-Manchester University NHS Foundation Trust (MFT)	0.023	0.028
2-Scotland	1-Queen Elizabeth University Hospital (QEH)	0.035	0.040
	2-Ninewells Hospital and Medical School (9W)	0.020	0.025
	3-Glasgow Royal Infirmary (GRI)	0.030	0.033
	4-Aberdeen Royal Infirmary (ARI)	0.025	0.028
	5-Royal Edinburgh Hospital (REH)	0.028	0.031
3-Wales	1-Ysbyty Gwynedd (YG)	0.015	0.020
	2-University Hospital of Wales (UHW)	0.012	0.015
	3-Morriston Hospital (MH)	0.010	0.012
	4-Royal Gwent Hospital (RGH)	0.008	0.010
	5-Wrexham Maelor Hospital (WMH)	0.007	0.008
4-Northern Ireland	1-Belfast City Hospital (BCH)	0.018	0.019
	2-Altnagelvin Area Hospital (AAH)	0.016	0.017
	3-Craigavon Area Hospital (CAH)	0.014	0.015
	4-South West Acute Hospital (SWAH)	0.012	0.013

Our primary goal is to devise a robust multilevel optimization framework that accurately emulates the intricacies of the COVID-19 vaccine allocation process, systematically addressing the specific priorities at each administrative tier. We meticulously elaborate on the hierarchical structure of the model, elucidating the decision variables at each level, formulating the corresponding objective functions, and systematically enumerating the constraints essential for a faithful representation of the vaccination planning landscape in the UK. Our formulated model aims to provide not only valuable insights but also strategic directives aimed at optimizing the distribution of COVID-19 vaccines within the intricate framework of LTCFs in the UK.

We present a multilevel optimization problem designed to simulate the vaccination allocation process, strategically organized across three distinct levels:

Level 1 – central management (government of the UK)

– **Objective** maximize the total number of COVID-19 vaccine doses allocated to the UK.

– **Possible decisions** x_{11} – number of vaccine doses allocated to the UK.

– **Objective function (level 1)** maximize x_{11} (vaccine doses allocated to the UK).

– **Constraints (level 1)** national vaccine production capacity: $x_{11} \leq 100$ million doses (assuming a procurement capacity of 100 million doses).

Level 2 – regional management (regions of the UK)

– **Objective** optimize vaccine distribution at the regional level based on population, infection rates, and healthcare infrastructure.
– **Possible decisions** x_{2i} represents the allocation of COVID-19 vaccine doses to National Health Service (NHS) region i.
– **Objective function (level 2)** maximize $\sum_{i=1}^{4} \lambda_i x_{2i}$ (vaccine utilization by region), where λ_i are positive coefficients with $\sum_{i=1}^{4} \lambda_i = 1$.

The objective at Level 2 is to optimize the distribution of COVID-19 vaccine doses across the four NHS regions, represented by the objective function $(x_{21}, x_{22}, x_{23}, x_{24})$. The coefficients λ_i reflect the priority assigned to each region, and they are determined based on the urgency and specific considerations set by the government.

– **Constraints (Level 2)**

- Regional population: $x_{2i} \leq P_i$ (total population of NHS region i).
- Regional infection rates: $x_{2i} \geq I_i$ (total number of confirmed COVID-19 cases in NHS region i).
- Regional healthcare infrastructure: $x_{2i} \geq C_i$ (total vaccine administration capacity of NHS region i).
- Vaccination allocation to regions constraint: $\sum_{i=1}^{4} x_{2i} \leq x_{11}$.

Level 3 – local management (hospitals within each region)

– **Objective** optimize vaccine allocation within each NHS region by prioritizing vulnerable populations and ensuring equitable access to vaccines.
– **Possible decisions** x_{3i}^{j} represents the number of COVID-19 vaccine doses allocated to NHS hospital j in NHS region i, where $j = 1, \ldots, j_i$.
– **Objective function (level 3)** maximize $\sum_{i=1}^{4} \sum_{j=1}^{j_i} \omega_i^j x_{3i}^j$, where $\omega_i^j = \lambda_i \xi_i^j$, subject to $\sum_{j=1}^{j_i} \xi_i^j = 1$ and ξ_i^j being positive coefficients.
The objective at Level 3 is to optimize the allocation of COVID-19 vaccine doses among NHS hospitals within each NHS hospital j in NHS region i, represented by the objective function:

$$\big(x_{31}^1, x_{31}^2, x_{31}^3, x_{31}^4, x_{31}^5, x_{31}^6, x_{31}^7, x_{31}^8, x_{32}^1, x_{32}^2, x_{32}^3, x_{32}^4, x_{32}^5, x_{33}^1, x_{33}^2, x_{33}^3, x_{33}^4, x_{33}^5, x_{34}^1, x_{34}^2, x_{34}^3, x_{34}^4\big).$$

The coefficients ω_i^j reflect the priority assigned to each NHS hospital j in NHS region i, and they are determined based on the urgency and specific considerations set by the government.

– **Constraints (level 3)**

- Care hospital capacity: $x_{3i}^j \leq B_{ij}$ (total number of beds in NHS hospital j in NHS region i).
This constraint ensures that the number of vaccine doses allocated to a hospital does not exceed its capacity to administer them based on the number of beds. This ensures efficient use of resources.
- Vulnerable populations: $x_{3i}^j \geq V_{ij}$ (number of individuals in vulnerable groups at NHS hospital j in NHS region i).
This constraint ensures that the number of vaccine doses allocated to a hospital is sufficient to cover at least the number of vulnerable individuals within its patient population. This prioritizes vulnerable groups.

TABLE 2. Matrix formulation of objective functions and constraints of the three levels programming problem.

Levels	Objective functions	Constraints
Level 1	$f_1(x) = x_{11}$	$x_{11} \leq 100$
Level 2	$f_2(x) = \sum_{i=1}^{4} \lambda_i x_{2i}$	$\begin{cases} \diamond\ x_{2i} \leq P_i & \Leftrightarrow \mathbf{Id}_4(\overline{x}^2)^T \leq \mathbf{P} \\ \diamond\ x_{2i} \geq I_i & \Leftrightarrow -\mathbf{Id}_4(\overline{x}^2)^T \leq -\mathbf{I} \\ \diamond\ x_{2i} \geq C_i & \Leftrightarrow -\mathbf{Id}_4(\overline{x}^2)^T \leq -\mathbf{C} \\ \diamond\ \sum_{i=1}^{4} x_{2i} \leq x_{11} & \Leftrightarrow \mathbf{1}_4(-\overline{x}^1, \overline{x}^2)^T \leq 0 \end{cases}$ *13 Constraints*
Level 3	$f_3(x) = \sum_{i=1}^{4} \sum_{j=1}^{j_i} \omega_i^j x_{3i}^j$	$\begin{cases} \diamond\ x_{3i}^j \leq B_{ij} & \Leftrightarrow \mathbf{Id}_7(\overline{x}^3)^T \leq \mathbf{B} \\ \diamond\ x_{3i}^j \geq V_{ij} & \Leftrightarrow -\mathbf{Id}_7(\overline{x}^3)^T \leq -\mathbf{V} \\ \diamond\ \sum_{i=1}^{4} \sum_{j=1}^{j_i} x_{3i}^j \leq \sum_{i=1}^{4} x_{2i} & \Leftrightarrow \mathbf{1}_{11}(-\overline{x}^2, \overline{x}^3)^T \leq 0 \end{cases}$ *19 Constraints*

- Vaccination allocation to hospitals constraint: $\sum_{i=1}^{4} \sum_{j=1}^{j_i} x_{3i}^j \leq \sum_{i=1}^{4} x_{2i}$ (total number of vaccine doses allocated to the four regions).
  This constraint ensures that the total number of vaccine doses allocated to hospitals in all regions does not exceed the total allocated to the UK. This maintains consistency and prevents exceeding the available supply.

Notations 6.1 (Tab. 2)**.**

- $\mathbf{Id}_4$ and $\mathbf{Id}_7$ are the identity matrices of order 4 and 7, respectively.
- $\mathbf{0}_{h_1 \times h_2}$ and $\mathbf{1}_{h_1 \times h_2}$: Matrices of order $h_1 \times h_2$, where $h_1, h_2 \in \mathbb{N}$, with all components equal to 0 and 1, respectively.
- Decision vector:

$$x := \left(\overline{x}^1, \overline{x}^2, \overline{x}^3\right) = \left(\underbrace{x_{11}}_{\overline{x}^1}, \underbrace{x_{21}, \ldots, x_{24}}_{\overline{x}^2}, \underbrace{x_{31}^1, \ldots, x_{31}^8, x_{32}^1, \ldots, x_{32}^5, x_{33}^1, \ldots, x_{33}^5, x_{34}^1, \ldots, x_{34}^4}_{\overline{x}^3} \right).$$

- Vectors: $\mathbf{P} = (P_i)$, $\mathbf{I} = (I_i)$, $\mathbf{C} = (C_i)$ (all of order 4), $\mathbf{B} = (B_{ij})$, $\mathbf{V} = (V_{ij})$ (both of order 7).

TABLE 3. Coefficients of objective functions f_2 and f_3 (scaled by 10^{-4}).

λ_i	i	j															
		1		2		3		4		5		6		7		8	
		ξ_i^j	ω_i^j	ξ_i^j	ω_i^j	ξ_i^j	ω_i^j	ξ_i^j	ω_i^j	ξ_i^j	ω_i^j	ξ_i^j	ω_i^j	ξ_i^j	ω_i^j	ξ_i^j	ω_i^j
0.4	1	0.12	0.05	0.14	0.06	0.10	0.04	0.25	0.08	0.13	0.09	0.12	0.05	0.15	0.06	0.22	0.07
0.3	2	0.25	0.08	0.16	0.10	0.21	0.13	0.18	0.06	0.20	0.12						
0.2	3	0.31	0.15	0.23	0.12	0.18	0.09	0.15	0.06	0.25	0.08						
0.1	4	0.30	0.03	0.27	0.02	0.23	0.07	0.20	0.02								

6.2. Numerical application

Consider the three-level monoobjective decentralized linear programming problem of the vaccination planning in long-term care facilities as follows:

$$\begin{aligned}
&\textbf{Level 1: } \max_{\overline{x}^1} f_1(x) = f_1\left(\overline{x}^1, \overline{x}^2, \overline{x}^3\right) = x_{11} \\
&\quad \text{such that } x_{21}, \ldots, x_{24}, \text{and } x_{31}^1, \ldots, x_{31}^8, x_{32}^1, \ldots, x_{32}^5, x_{33}^1, \ldots, x_{33}^5, x_{34}^1, \ldots, x_{34}^4 \text{ solves:} \\
&\textbf{Level 2: } \max_{\overline{x}^2} f_2(x) = f_2\left(\overline{x}^1, \overline{x}^2, \overline{x}^3\right) = \lambda_1 x_{21} + \lambda_2 x_{22} + \lambda_3 x_{23} + \lambda_4 x_{24} \\
&\quad \text{such that } x_{31}^1, \ldots, x_{31}^8, x_{32}^1, \ldots, x_{32}^5, x_{33}^1, \ldots, x_{33}^5, x_{34}^1, \ldots, x_{34}^4 \text{ solves:} \\
&\textbf{Level 3: } \max_{\overline{x}^3} f_3(x) = f_3\left(\overline{x}^1, \overline{x}^2, \overline{x}^3\right) = \sum_{i=1}^{4} \sum_{j=1}^{j_i} \omega_i^j x_{3i}^j,
\end{aligned} \tag{27}$$

subject to:

$$x \in S = \left\{x \in \mathbb{R}^{27} \ : \ Ax = b, \ x \geq 0\right\} \neq \emptyset,$$

where A is a matrix of order 29×27 and b is a vector of order 29, defined as follows:

$$A = \begin{pmatrix} 1 & \mathbf{0}_{1\times 4} & \mathbf{0}_{1\times 7} \\ \mathbf{0}_{4\times 1} & \mathbf{Id}_4 & \mathbf{0}_{4\times 7} \\ \mathbf{0}_{4\times 1} & -\mathbf{Id}_4 & \mathbf{0}_{4\times 7} \\ \mathbf{0}_{4\times 1} & -\mathbf{Id}_4 & \mathbf{0}_{4\times 7} \\ -1 & \mathbf{1}_{1\times 4} & \mathbf{0}_{1\times 7} \\ \mathbf{0}_{7\times 1} & \mathbf{0}_{7\times 4} & \mathbf{Id}_7 \\ \mathbf{0}_{7\times 1} & \mathbf{0}_{7\times 4} & -\mathbf{Id}_7 \\ 0 & -\mathbf{1}_{1\times 4} & \mathbf{1}_{1\times 7} \end{pmatrix}, \quad b = \begin{pmatrix} 100 \\ \mathbf{P} \\ -\mathbf{I} \\ -\mathbf{C} \\ 0 \\ \mathbf{B} \\ -\mathbf{V} \\ 0 \end{pmatrix}.$$

Assume the government has selected the parameters based on the importance of care hospital capacity **B** (Tab. 3):

Using the monocriteria primal simplex method, we maximize each of the three objective functions, f_1, f_2, and f_3. The results of this optimization process are presented in Table 4.

Applying our Algorithm 3, given that we have three DMs (3 levels, $P = 3$), we will undergo two iterations of the while loop in step 3. The DM_1 and DM_2 will, in turn, narrow down the intervals of the components under their control. The compromise will be reached by DM_3, and the algorithm terminates.

Let α_1 and α_2 denote the parameters chosen by the first two DMs to reduce the respective intervals of the variables under their control in the case where $\overset{o}{x}{}_{p-1,j}^{(p-1)} = l_{p-1,j}^{(p-1)}$ or $\overset{o}{x}{}_{p-1,j}^{(p-1)} = u_{p-1,j}^{(p-1)}$, as follows:

$$\left[l_{p,j}^{(p)}, u_{p,j}^{(p)}\right] = \begin{cases} \left[l_{p-1,j}^{(p-1)} + \alpha_{p-1}\left(u_{p-1,j}^{(p-1)} - l_{p-1,j}^{(p-1)}\right), u_{p-1,j}^{(p-1)}\right], & \text{if } l_{p-1,j}^{(p-1)} = \overset{o}{x}{}_{p-1,j}^{(p-1)}, \\ \left[l_{p-1,j}^{(p-1)}, u_{p-1,j}^{(p-1)} - \alpha_{p-1}\left(u_{p-1,j}^{(p-1)} - l_{p-1,j}^{(p-1)}\right)\right], & \text{if } u_{p-1,j}^{(p-1)} = \overset{o}{x}{}_{p-1,j}^{(p-1)}, \end{cases} \begin{array}{l} p = 2, 3, \\ j = 1, \ldots, n_{p-1} \\ 0 < \alpha_{p-1} < 1. \end{array}$$

The compromise solutions $\overset{c}{x}$ are then obtained as functions of α_1 and α_2, as detailed in Table 5.

TABLE 4. Optimal solutions $\overset{o}{x}_1$, $\overset{o}{x}_2$, $\overset{o}{x}_3$ and optimal values $f_1^{\max}$, $f_2^{\max}$, $f_3^{\max}$ (scaled by 10^{-4}).

DMs	$f_p^{\max}$	$\overset{o}{x}_{11}$	$\overset{o}{x}_{21}$	$\overset{o}{x}_{22}$	$\overset{o}{x}_{23}$	$\overset{o}{x}_{24}$	$\overset{o}{x}{}^1_{31}$	$\overset{o}{x}{}^2_{31}$	$\overset{o}{x}{}^3_{31}$	$\overset{o}{x}{}^4_{31}$	$\overset{o}{x}{}^5_{31}$	$\overset{o}{x}{}^6_{31}$	$\overset{o}{x}{}^7_{31}$	$\overset{o}{x}{}^8_{31}$
1	10 000	10 000	5180	524	302	183	3	3	2	3	3	3	4	2
2	2505	7289.6	5648	545	316	191	2.6	3.1	2.1	2.9	3.3	2.8	3.6	2.4
3	2762.6	699.1	5147.5	524.7	302.7	183.6	3	3.5	2.5	3.2	3.4	3	3.8	2.8

DMs	$\overset{o}{x}{}^1_{32}$	$\overset{o}{x}{}^2_{32}$	$\overset{o}{x}{}^3_{32}$	$\overset{o}{x}{}^4_{32}$	$\overset{o}{x}{}^5_{32}$	$\overset{o}{x}{}^1_{33}$	$\overset{o}{x}{}^2_{33}$	$\overset{o}{x}{}^3_{33}$	$\overset{o}{x}{}^4_{33}$	$\overset{o}{x}{}^5_{33}$	$\overset{o}{x}{}^1_{34}$	$\overset{o}{x}{}^2_{34}$	$\overset{o}{x}{}^3_{34}$	$\overset{o}{x}{}^4_{34}$
1	4	2	3	3	3	2	1	1	1	1	2	2	1	1
2	3.6	2.1	3.1	2.6	2.9	1.6	1.3	1.1	0.9	0.7	1.8	1.6	1.4	1.2
3	4	2.5	3.3	2.8	3.1	2	1.5	1.2	1	0.8	1.9	1.7	1.5	1.3

TABLE 5. Comprehensive overview of solutions for the linear three-level problem (27) obtained through primal simplex method and Gabasov's adaptive method across various α_1 and α_2 values (scaled by 10^{-3}).

		Simplex method						Adaptive method																													
		$\overset{c}{x}_{11}$	$\overset{c}{x}_{21}$	$\overset{c}{x}_{22}$	$\overset{c}{x}_{23}$	$\overset{c}{x}_{24}$	$\overset{c}{x}_{11}$	$\overset{c}{x}_{21}$	$\overset{c}{x}_{22}$	$\overset{c}{x}_{23}$	$\overset{c}{x}_{24}$	$\overset{c1}{x}_{31}$	$\overset{c2}{x}_{31}$	$\overset{c3}{x}_{31}$	$\overset{c4}{x}_{31}$	$\overset{c5}{x}_{31}$	$\overset{c6}{x}_{31}$	$\overset{c7}{x}_{31}$	$\overset{c8}{x}_{31}$	$\overset{c1}{x}_{32}$	$\overset{c2}{x}_{32}$	$\overset{c3}{x}_{32}$	$\overset{c4}{x}_{32}$	$\overset{c5}{x}_{32}$	$\overset{c1}{x}_{33}$	$\overset{c2}{x}_{33}$	$\overset{c3}{x}_{33}$	$\overset{c4}{x}_{33}$	$\overset{c5}{x}_{33}$	$\overset{c1}{x}_{34}$	$\overset{c2}{x}_{34}$	$\overset{c3}{x}_{34}$	$\overset{c4}{x}_{34}$
α_1	α_2	Adaptive method and simplex method																																	
	0	83 046	54 423	5323	3070	1854	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0.25	83 046	54 423	5323	3070	1854	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
0	0.5	83 046	54 423	5323	3070	1854	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0.75	83 046	54 423	5323	3070	1854	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	1	83 046	54 423	5323	3070	1854	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0	81 552	54 376	5323	3070	1854	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0.25	81 487	54 064	5304	3057	1853	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
0.25	0.5	81 382	53 502	5279	3042	1851	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0.75	81 199	52 453	5259	3033	1846	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	1	81 029	51 475	5245	3024	1855	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0	79 557	54 295	5324	3071	1854	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0.25	79 505	54 004	5306	3071	1854	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
0.5	0.5	79 437	53 496	5285	3072	1855	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0.75	79 268	52 491	5260	3060	1847	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	1	78 905	51 475	5245	3056	1854	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0	76 778	54 168	5322	3071	1854	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0.25	76 725	53 897	5304	3058	1847	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
0.75	0.5	76 657	53 425	5283	3044	1839	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0.75	76 589	52 501	5263	3032	1838	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	1	75 794	51 475	5245	3024	1833	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0	69 910	54 073	5305	3068	1863	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0.25	69 910	54 073	5305	3068	1863	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
1	0.5	69 910	54 073	5305	3068	1863	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	0.75	69 910	54 073	5305	3068	1863	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13
	1	69 910	54 073	5305	3068	1863	69 910	51 475	5245	3024	1833	30	35	25	32	34	30	38	28	40	25	33	28	31	20	15	12	10	08	19	17	15	13

6.2.1. Table 6 explanation: comparative analysis of primal simplex and Gabasov's adaptive methods

Within the specified range of $[0, 1]$, the parameters α_1 and α_2 are carefully chosen with a subdivision step of $[0, 1]$, set at 0.25. Subsequently, a comprehensive exploration of all possible combinations of α_1 and α_2 is conducted by the two DMs, leading to the data presented in Table 6. The performance metrics under scrutiny include time efficiency, memory consumption, and objective function values. The computational resolution is executed using Matlab 2007b on a PC featuring an "Intel(R) Celeron(R) CPU N3150 @ 1.60 GHz 1.60 GHz" processor with 4 GB of RAM. Time is measured in seconds, while memory is recorded in megabytes (Mb). Notably, the objective function values $\left(\overset{c}{f}_1, \overset{c}{f}_2, \overset{c}{f}_3\right)$ are scaled by a factor of 10^{-4} for improved readability. These objective function values signify the maximum outcomes attained by each method concerning the respective objective functions, computed over the compromise solutions $\overset{c}{x}$.

TABLE 6. Results of the primal simplex and Gabasov's adaptive methods for varying α_1 and α_2. The values in the "Values" columns are scaled by 10^{-4}.

		Primal simplex method					Gabasov's adaptive method				
α_1	α_2	Time (s)	Memory (Mb)	Values ($\times 10^{-4}$) $\overset{c}{f}_1$	$\overset{c}{f}_2$	$\overset{c}{f}_3$	Time (s)	Memory (Mb)	Values ($\times 10^{-4}$) $\overset{c}{f}_1$	$\overset{c}{f}_2$	$\overset{c}{f}_3$
0	0	0.9751	105 572	8304.6	2416.5	2.7626	1.2137	96 002	6991	2295.1	2.7626
	0.25	0.9491	49 298	8304.6	2416.5	2.7626	0.9276	59 340	6991	2295.1	2.7626
	0.5	0.9507	16 576	8304.6	2416.5	2.7626	0.9162	16 576	6991	2295.1	2.7626
	0.75	0.9564	16 576	8304.6	2416.5	2.7626	0.9139	16 576	6991	2295.1	2.7626
	1	0.9639	16 576	8304.6	2416.5	2.7626	0.9215	16 576	6991	2295.1	2.7626
0.25	0	0.9534	16 616	8155.2	2416.5	2.7626	0.9206	16 616	6991	2295.1	2.7626
	0.25	0.9600	16 576	8148.7	2401.3	2.7626	0.9177	16 576	6991	2295.1	2.7626
	0.5	0.9596	16 576	8138.2	2377.8	2.7626	0.9186	16 576	6991	2295.1	2.7626
	0.75	0.9532	16 576	8119.9	2335.0	2.7626	0.9146	16 576	6991	2295.1	2.7626
	1	0.9658	16 576	8102.9	2295.4	2.7626	0.9160	16 576	6991	2295.1	2.7626
0.5	0	0.9500	16 616	7955.7	2411.5	2.7626	0.9201	16 616	6991	2295.1	2.7626
	0.25	0.9553	16 576	7950.5	2399.3	2.7626	0.9175	16 576	6991	2295.1	2.7626
	0.5	0.9582	16 576	7943.7	2378.3	2.7626	0.9205	16 576	6991	2295.1	2.7626
	0.75	0.9597	16 576	7926.8	2337.1	2.7626	0.9194	16 576	6991	2295.1	2.7626
	1	0.9641	16 576	7890.5	2296.0	2.7626	0.9185	16 576	6991	2295.1	2.7626
0.75	0	0.9581	16 616	7677.8	2406.3	2.7626	0.9212	16 616	6991	2295.1	2.7626
	0.25	0.9604	16 576	7672.5	2394.6	2.7626	0.9212	16 576	6991	2295.1	2.7626
	0.5	0.9532	16 576	7665.7	2374.8	2.7626	0.9256	16 576	6991	2295.1	2.7626
	0.75	0.9592	16 576	7658.9	2337.0	2.7626	0.9178	16 576	6991	2295.1	2.7626
	1	0.9825	16 576	7579.4	2295.1	2.7626	0.9296	16 576	6991	2295.1	2.7626
1	0	0.9534	16 616	6991	2402.1	2.7626	0.9222	16 616	6991	22951	2.7626
	0.25	0.9603	16 576	6991	2402.1	2.7626	0.9158	16 576	6991	22951	2.7626
	0.5	0.9556	16 576	6991	2402.1	2.7626	0.9191	16 576	6991	22951	2.7626
	0.75	0.9688	16 576	6991	2402.1	2.7626	0.9317	16 576	6991	22951	2.7626
	1	0.9601	16 576	6991	2402.1	2.7626	0.9169	16 576	6991	22951	2.7626

6.3. Discussion

6.3.1. Hospitals and care capacity

As per information from the NHS website, the UK boasts around 2,500 hospitals. The data presented in Table 1 corresponds to 22 of these hospitals, constituting approximately 0.88% of the total. It's crucial to note that the table's coverage is restricted to hospitals with available data regarding vulnerable populations and care capacity. Consequently, the actual count of hospital beds in the UK might surpass the provided figure, and the 22 hospitals may represent only a fraction of the total care capacity. As a result, pinpointing the precise percentage of the UK's total care capacity represented by the hospitals in the table remains challenging, but it is likely to be less than 0.88%.

6.3.2. Vaccine planning in LTCFs

The mathematical model outlined in Section 6.1, as employed in the vaccine planning example, provides a simplified representation of the real-world problem. Notably, it does not account for all factors influencing vaccine distribution, such as the prevailing epidemiological conditions, vaccination policies, and resource availability. The model serves primarily an illustrative purpose, demonstrating the application of our method to vaccine planning in LTCFs. Its aim is to highlight how our approach can be used to design effective and equitable distribution

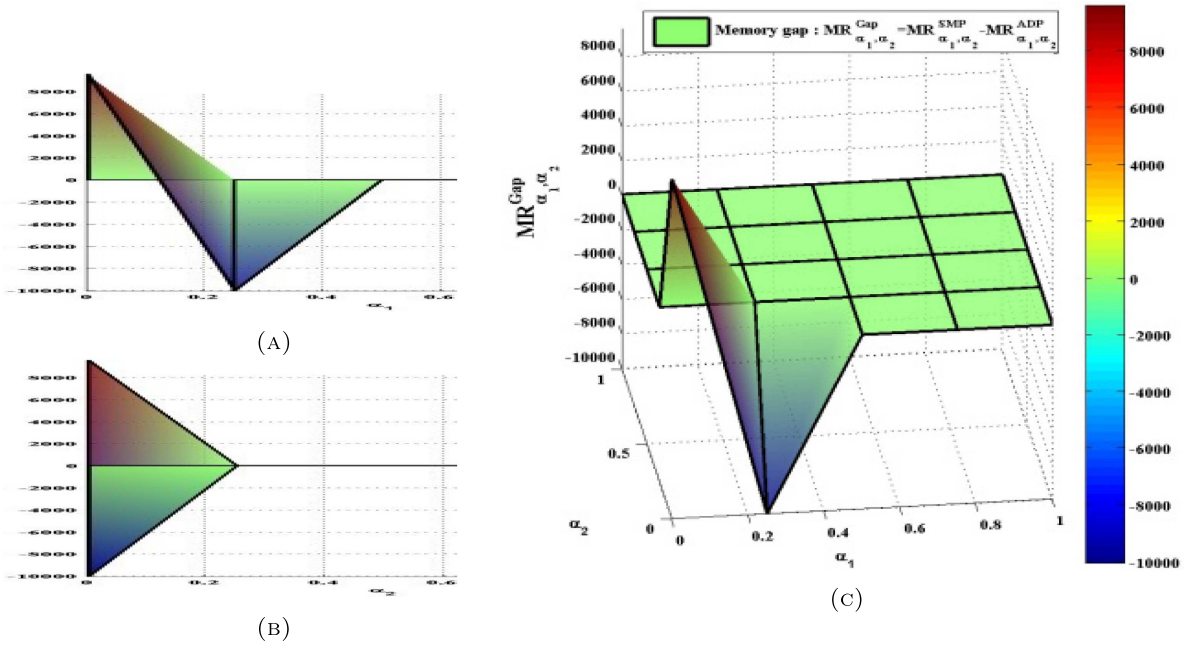


FIGURE 3. Memory allocation variability: a comparative study between primal simplex and Gabasov's adaptive methods, influenced by α_1 and α_2. (A) Memory allocation at $\alpha_2 = 0$. (B) Memory allocation at $\alpha_1 = 0$. (C) Spatial insights into memory allocation: contrasting primal simplex and Gabasov's adaptive method. **Memory utilization discrepancy:** the variable $MR^{\text{Gap}}_{\alpha_1,\alpha_2}$ characterizes the difference in memory usage between the Gabasov's adaptive (ADP) method and the primal simplex (SMP) method. In this context, $MR^{\text{ADP}}_{\alpha_1,\alpha_2}$ represents the memory consumption of the ADP method under parameters α_1 and α_2, while $MR^{\text{SMP}}_{\alpha_1,\alpha_2}$ represents the memory consumption of the SMP method under the same parameters.

plans in real-world contexts, acknowledging the need for further refinement to address additional complexities in practical scenarios.

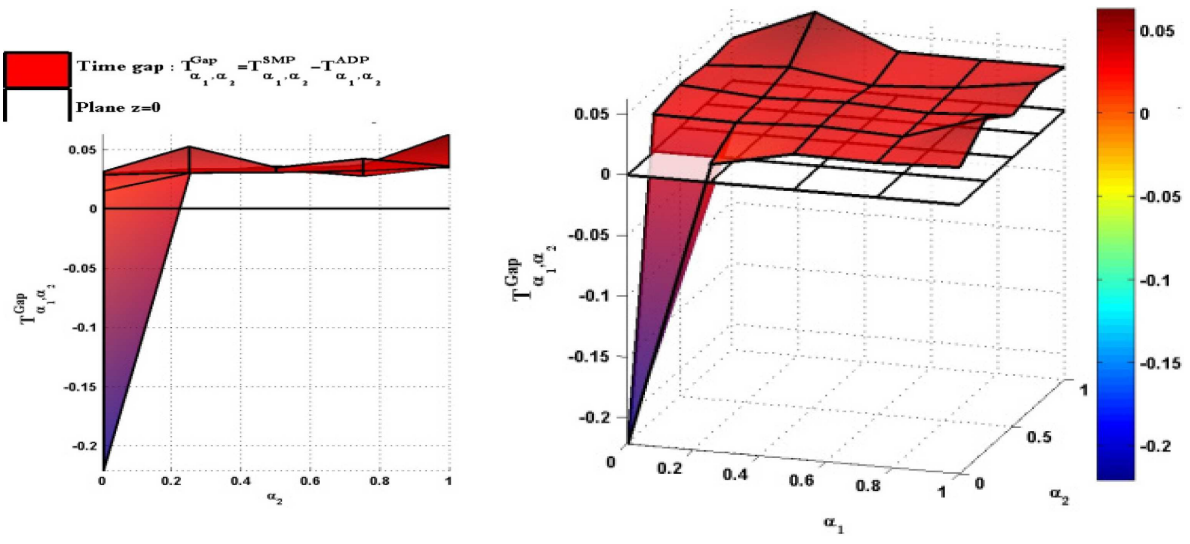
To refine the model, various enhancements can be contemplated, such as updating it to align with the present epidemiological landscape, adjusting it to accommodate evolving vaccination policies, and broadening its scope to encompass additional variables influencing vaccine distribution. While these enhancements promise a more precise depiction of the intricate facets of vaccine planning, it's essential to acknowledge that they may result in a non-linear model. Consequently, this non-linearity poses a challenge in applying our proposed method in such cases.

6.3.3. Discussion on optimization results

The results presented in Table 6 illuminate the comparative performance of the primal simplex and Gabasov's adaptive methods under varying parameter configurations (α_1 and α_2). Noticeable trends in time and memory consumption emerge as these parameters change, revealing a consistent pattern between the two methods.

To enhance the interpretation of the results in Table 6, visual aids are provided through the graphs in Figures 3–5. These graphical representations offer a clearer and more intuitive understanding of the outcomes, contributing to a comprehensive analysis and facilitating a nuanced examination of the data.

Consideration of the computational environment is crucial in interpreting the obtained results. The utilization of Matlab 2007b on a computer with "Intel(R) Celeron(R) CPU N3150 @ 1.60 GHz 1.60 GHz" processor and



(A) (B)

FIGURE 4. Comparison of temporal discrepancies between the primal simplex *vs.* Gabasov's adaptive method based on α_1 and α_2 parameters. (A) Planar insights into primal simplex *vs.* Gabasov's adaptive method dynamics at $\alpha_1 = 0$. (B) Temporal efficiency analysis: a 3D exploration of the primal simplex *vs.* Gabasov's adaptive approach. **Temporal disparity analysis:** the variable $T^{\text{Gap}}_{\alpha_1,\alpha_2}$ delineates the temporal deviation between the ADP and the SMP methods. Here, $T^{\text{ADP}}_{\alpha_1,\alpha_2}$ represents the temporal consumption of the ADP method under parameters α_1 and α_2, while $T^{\text{SMP}}_{\alpha_1,\alpha_2}$ corresponds to the temporal consumption of the SMP method under the same parameters.

4 GB of RAM – indicative of lower-end performance and an outdated Matlab version, inevitably influences the execution time and potentially the precision of our method. Recognizing these limitations is paramount. Upgrading to a more powerful system with modern hardware specifications and an updated Matlab version is expected to significantly improve result quality, especially in terms of reduced execution time. This emphasizes the importance of considering computational resources in interpreting and generalizing our study's findings.

For the scenario where $\alpha_1 = 0$ and $\alpha_2 = 0$, representing DMs making no reduction to their intervals (maximizing flexibility for followers), the Gabasov's adaptive method exhibits a longer execution time but requires less memory compared to the primal simplex method. However, for other values of α_1 (ranging from 0 to 1) and α_2 (ranging from 0.25 to 1), the Gabasov's adaptive method consistently demonstrates reduced time consumption compared to the primal simplex method. Regarding memory usage, the Gabasov's adaptive method consumes more memory than the primal simplex method only for $\alpha_1 = 0$ and $\alpha_2 = 0.25$. Subsequently, both methods consume the same amount of memory for the remaining values. This observation is reflected in the peaks in the memory graph 3 and the negative spike in the time graph 4, followed by stabilization at positive values, indicating an advantage in favor of the Gabasov's adaptive method. These results underscore the significance of parameter configuration in determining the relative performance of the two methods.

An intriguing perspective is to explain why this influence particularly manifests itself in the neighborhood of the values $\alpha_1 = \alpha_2 = 0$.

These insights are pivotal for practitioners and researchers in selecting an optimization approach based on specific requirements and resource constraints. The results contribute to a nuanced understanding of the trade-offs

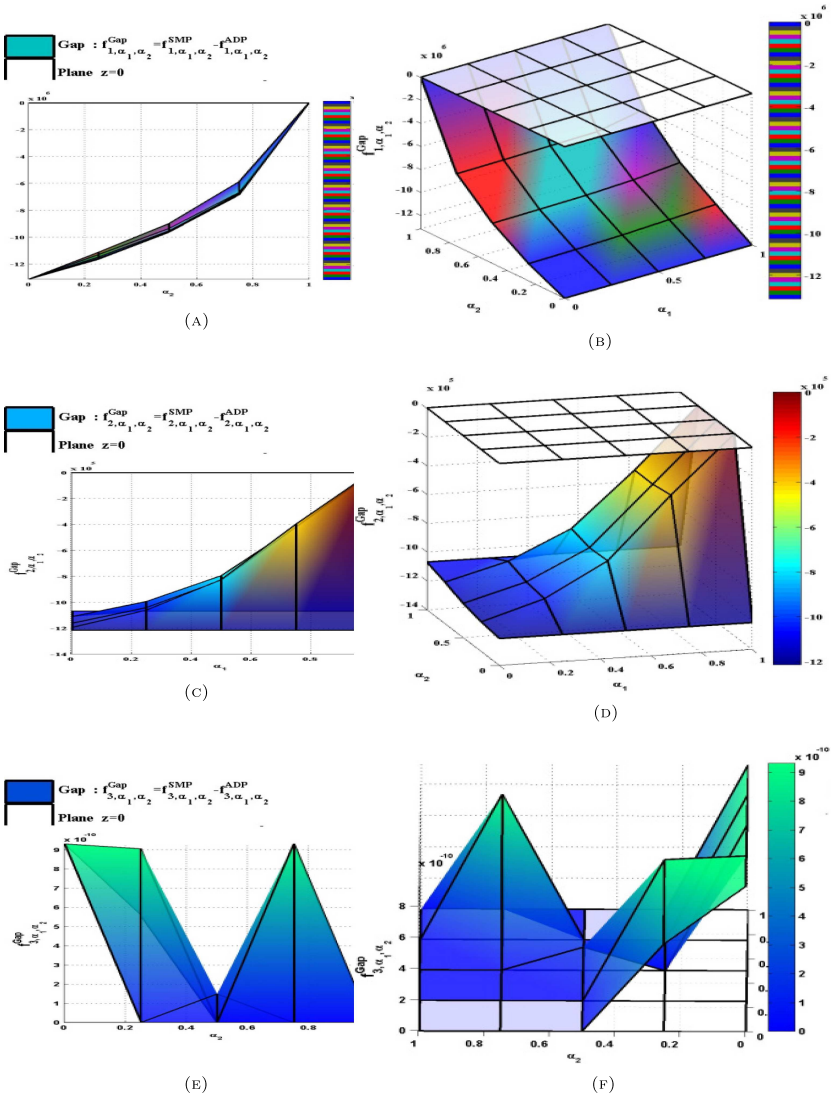


FIGURE 5. Divergence in objective function values: comparative analysis of primal simplex and Gabasov's adaptive methods with respect to α_1 and α_2. (A) Snapshot at $\alpha_1 = 0$: maximizing f_1 – primal simplex *vs.* Gabasov's adaptive method. (B) Solution peaks for f_1: primal simplex *vs.* Gabasov's adaptive method (variations in α_1 and α_2). (C) Visualizing $\alpha_2 = 0$: peak values in f_2 – primal simplex *vs.* Gabasov's adaptive method (variations in α_1 and α_2). (D) Maximums in f_2: comparative study between primal simplex and Gabasov's adaptive method (different α_1 and α_2). (E) Snapshot at $\alpha_1 = 0$: maximizing f_3 – primal simplex vs. Gabasov's adaptive method. (F) Optimal peaks in f_3: analyzing the difference between primal simplex and Gabasov's adaptive method (varied α_1 and α_2). **Objective function disparity:** the variable $f^{\text{Gap}}_{p,\alpha_1,\alpha_2}$ signifies the difference in the maximum values achieved using the SMP method and the ADP method, where $p = 1, 2, 3$. Here, $f^{\text{ADP}}_{p,\alpha_1,\alpha_2}$ and $f^{\text{SMP}}_{p,\alpha_1,\alpha_2}$ respectively denote the maximum values attained by the ADP and SMP methods' solutions under parameters α_1 and α_2.

between the primal simplex and Gabasov's adaptive methods, guiding the selection of the most suitable method based on the optimization problem's characteristics and the available computational resources.

7. Conclusion

This study pursued a dual objective: firstly, the application of the Gabasov's adaptive method to solve a ML-MO-DLPPs; secondly, the refinement of Sinha and Sinha's algorithm [28] by replacing the primal simplex method with the Gabasov's adaptive method in step 10. An additional innovation was the restructuring of steps 3 to 9 using the interval reduction map ξ, consolidating these eight steps into a single step (Step 3 in our algorithm).

By harnessing the constructive Gabasov's adaptive method, we introduced a detailed algorithm for addressing ML-MO-DLPPs, with the establishment of the interval reduction map playing a pivotal role. The convergence of our algorithms is evident from the results of the Gabasov's adaptive method, and a numerical example was provided to illustrate the efficacy of our algorithm.

The discussions in the preceding sections have illuminated the practical implications of our work. While our algorithm exhibits enhanced efficiency in solving ML-MO-DLPP, it is essential to recognize both its scope and limitations. The comparisons in Table 6 offer informations on the computational performance of the primal simplex and Gabasov's adaptive methods under different parameter configurations (α_1 and α_2). These insights serve as a guide for selecting the most suitable optimization approach based on specific requirements and resource constraints.

In conclusion, our contributions extend beyond algorithmic enhancements. We have not only presented a comprehensive understanding of the trade-offs between optimization methods but also discussed their implications for real-world problem-solving, exemplified through a numerical example involving the distribution of COVID-19 vaccines in LTCFs. This application, specifically in healthcare planning, not only advances the field of optimization but also encourages future research endeavors that consider the broader context of application.

Conflicts of interest

The author declare that there is no conflict of interest regarding the publication of this manuscript.

Data availability statement

The data utilized in this study is sourced from various official entities and providers, ensuring the reliability and credibility of the information. Links to the primary data sources are provided below:

- Population data: Office for National Statistics (ONS), National Records of Scotland (NRS), Welsh Government, Northern Ireland Statistics and Research Agency (NISRA).
- Cases data: Public Health England, Public Health Scotland, Public Health Wales, and the Public Health Agency (Northern Ireland).
- Administration capacity data: National Health Service (NHS) England, Scottish Government, Welsh Government, Department of Health (Northern Ireland).
- Equitable vaccine distribution target: World Health Organization (WHO), United Nations Children's Fund (UNICEF).
- V_{ij} values were estimated by ONS, NRS, Welsh government, and NISRA.
- B_{ij} values were provided by the respective hospitals.

It should be noted that Google Bard provided assistance in accessing and collating all the data utilized in this study.

References

[1] M.A. Abo-sinna, Pareto optimality for bi-level programming problem with fuzzy parameters. *Opsearch* **38** (2001) 372–393.

[2] M.A. Abo-Sinna and I.A. Baky, Interactive balance space approach for solving multi-level multi-objective programming problems. *Inf. Sci.* **177** (2007) 3397–3410.

[3] F. Ahmad, S. Ahmad, A.T. Soliman and M. Abdollahian, Solving multi-level multiobjective fractional programming problem with rough interval parameter in neutrosophic environment. *RAIRO-Oper. Res.* **55** (2021) 2567–2581.

[4] N. Aydin and Z. Cetinkale, Analyses on ICU and non-ICU capacity of government hospitals during the COVID-19 outbreak via multi-objective linear programming: an evidence from Istanbul. *Comput. Biol. Med.* **146** (2022) 105562.
[5] E. Babaee Tirkolaee, H. Golpîra, A. Javanmardan and R. Maihami, A socio-economic optimization model for blood supply chain network design during the COVID-19 pandemic: an interactive possibilistic programming approach for a real case study. *Soc.-Econ. Planning Sci.* **85** (2023) 101439.
[6] I.A. Baky, Fuzzy goal programming algorithm for solving decentralized bi-level multi-objective programming problems. *Fuzzy Sets Syst.* **160** (2009) 2701–2713.
[7] I.A. Baky, Solving multi-level multi-objective linear programming problems through fuzzy goal programming approach. *Appl. Math. Modell.* **34** (2010) 2377–2387.
[8] I.A. Baky, M.H. Eid and M.A. El Sayed, Bi-level multi-objective programming problem with fuzzy demands: a fuzzy goal programming algorithm. *Opsearch* **51** (2014) 280–296.
[9] O. Ben-Ayed, Bilevel linear programming. *Comput. Oper. Res.* **20** (1993) 485–501.
[10] J. Bracken and J.T. McGill, Mathematical programs with optimization problems in the constraints. *Oper. Res.* **21** (1973) 37–44.
[11] J. Bracken and J.T. McGill, Technical note – a method for solving mathematical programs with nonlinear programs in the constraints. *Oper. Res.* **22** (1974) 917–1133.
[12] B. Colson, P. Marcotte and G. Savard, Bilevel programming: a survey. *4OR* **3** (2005) 87–107.
[13] R. Gabasov and F.M. Kirillova, Linear Programming Methods. House, Minsk (1978).
[14] R. Gabasov and F.M. Kirillova, New linear programming methods and their application to optimal control. *IFAC Proc. Vol.* **12** (1979) 17–30.
[15] R. Gabasov, F.M. Kirillova and S.V. Prischepova, Optimal Feedback Control. *Lecture Notes in Control and Information Sciences*. Springer, Berlin, Heidelberg (1995).
[16] M. Kaci and S. Radjef, The set of all the possible compromises of a multi-level multi-objective linear programming problem. *Croatian Oper. Res. Rev.* **13** (2022) 13–30.
[17] M. Kaci and S. Radjef, An adaptive method to solve multilevel multiobjective linear programming problems. *Oper. Res. Decisions* **33** (2023) 29–44.
[18] V.V. Kalashnikov, S. Dempe, G.A. Pérez-Valdés, N.I. Kalashnykova and J.F. Camacho-Vallejo, Bilevel programming and applications. *Math. Prob. Eng.* **2015** (2015) 1–17.
[19] K. Lachhwani, On solving multi-level multi objective linear programming problems through fuzzy goal programming approach. *Opsearch* **51** (2014) 624–637.
[20] K. Lachhwani, Solving the general fully neutrosophic multi-level multiobjective linear programming problems. *Opsearch* **58** (2021) 1192–1216.
[21] A. Miniguano-Trujillo, F. Salazar, R. Torres, P. Arias and K. Sotomayor, An integer programming model to assign patients based on mental health impact for tele-psychotherapy intervention during the COVID-19 emergency. *Health Care Manag. Sci.* **24** (2021) 286–304.
[22] D. Mollalign, A. Mushi and B. Guta, Solving multiobjective multilevel programming problems using two-phase intuitionistic fuzzy goal programming method. *J. Comput. Sci.* **63** (2022) 1877–7503.
[23] S. Nayak and A. Ojha, On multi-level multi-objective linear fractional programming problem with interval parameters. *RAIRO-Oper. Res.* **53** (2019) 1601–1616.
[24] S. Pardeshi, S. Gawade and P. Hemant, Student learning time analysis during COVID-19 using linear programming – simplex method. *Soc. Sci. Human. Open* **5** (2022) 100266.
[25] C.O. Pieume, P. Marcotte, L.P. Fotso and P. Siarry, Solving bilevel linear multiobjective programming problems. *Am. J. Oper. Res.* **01** (2011) 214–219.
[26] M. Sakawa and I. Nishizaki, Interactive fuzzy programming for multi-level programming problems: a review. *Int. J. Multicriteria Decis. Making* **2** (2012) 241–266.
[27] S. Sinha, Fuzzy programming approach to multi-level programming problems. *Fuzzy Sets Syst.* **136** (2003) 189–202.
[28] S.B. Sinha and S. Sinha, A linear programming approach for linear multi-level programming problems. *J. Oper. Res. Soc.* **55** (2004) 312–316.
[29] M. Tavana, K. Govindan, A.K. Nasr, M.S. Heidary and H. Mina, A mathematical programming approach for equitable COVID-19 vaccine distribution in developing countries. *Ann. Oper. Res.* (2021).

[30] L. Vicente and P. Calamai, Bilevel and multilevel programming: a bibliography review. *J. Global Optim.* **5** (1994) 291–306.