VIETNAM NATIONAL UNIVERSITY, HANOI

INTERNATIONAL SCHOOL

RESEARCH REPORT

Optimal repatriation scheduling for Vietnam: A MILP case study of the COVID-19 campaign

Course: INS3048 - Optimization in Quantitative Management

Name: Quan - Manh NGUYEN

Student ID: 21070555

Class: BDA2021B

Cohort: QH-2021Q

Supervisor: Dr. Thuan-Quang NGUYEN

June 13, 2025

Declaration of authorship

I, Linh - Gia PHAN, declare that this report, titled "*Optimization of Repatriation Scheduling: A Mixed Integer Linear Programming Approach Applied to the Vietnamese Context during the COVID-19 Pandemic*", and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree at this University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this report is entirely my own work.
- I have acknowledged all main sources of help.

Acknowledgements

I would like to express my sincere gratitude to my course lecturer, Dr. Thuan - Quang NGUYEN, for his invaluable guidance, insightful feedback, and unwavering support throughout the duration of this project. Their expertise in optimization significantly shaped the direction and quality of this research.

I also acknowledge the AI assistant(Gemini Deep Research 2.5 Pro) for its contribution in compiling the initial background report on Vietnam's repatriation efforts, which provided a foundational dataset for this study.

Finally, I am grateful to my family and friends for their encouragement and understanding.

Abstract

This study addresses the Repatriation Scheduling Problem (RSP) by applying a Mixed Integer Linear Programming (MILP) model to the specific context of Vietnam's 2020-2022 citizen repatriation campaign. A key contribution of this research is the development of a "Vietnamized" dataset, which synthesizes national resource constraints and humanitarian priorities to tailor the general model proposed by Al-Shihabi & Mladenović (2022). Implemented in Python with PuLP, the model yielded an optimal solution facilitating the repatriation of 2,211 high-priority citizens, achieving a total weighted score of 22,110. A critical insight from the analysis is that transportation capacity, rather than quarantine facilities, was the predominant system bottleneck. This work demonstrates the MILP model's efficacy as a strategic tool for optimizing resource allocation in complex, real-world humanitarian logistics scenarios, providing a tangible framework for future crisis management.

Keywords: Repatriation Scheduling Problem (RSP), Mixed Integer Linear Programming (MILP), humanitarian Logistics, optimization, COVID-19, resource allocation.

iii

Contents

List of Tables

Chapter 1

Introduction

1.1 Background & motivation

1.1.1 The impact of COVID-19 on international travel & repatriation needs

The onset of the COVID-19 pandemic in early 2020 triggered an unprecedented and abrupt shutdown of global travel. As nations worldwide sealed their borders and suspended commercial flights to curb the virus's spread, a severe humanitarian crisis unfolded: millions of individuals, including tourists, students, and workers, were suddenly stranded abroad. They were confronted with a precarious existence, grappling with expiring visas, financial hardship, loss of accommodation, and uncertain access to healthcare, all while being separated from their families during a global health emergency.

This situation created a profound dilemma for governments, forcing them to balance their duty to protect citizens abroad with the critical need to prevent the importation of the virus. The only viable solution was the launch of massive, state-led repatriation campaigns. These operations were fundamentally different from commercial aviation; they were complex humanitarian missions defined not by profit, but by urgent need and severe resource constraints. It is from this high-stakes context of widespread disruption and limited capacity that the Repatriation Scheduling Problem (RSP) emerges as a critical challenge for strategic optimization.

1.1.2 Vietnam's national repatriation campaign (2020-2022)

In response to the crisis that left tens of thousands of its citizens stranded abroad, the Government of Vietnam initiated a large-scale repatriation campaign, framed as a humanitarian effort to prioritize the return of its most vulnerable individuals. The scale of demand was immense—with up to 50,000 registrations in Japan alone, for example—and consistently outstripped the available capacity of flights and quarantine facilities. This significant imbalance between demand and resources created a complex scheduling challenge.

Operationally, the campaign was a complex, multi-agency endeavor managed through Vietnam's diplomatic missions abroad, which served as the central hubs for receiving applications. These missions coordinated with key domestic ministries and national airlines to arrange the specialized flights. This structured, state-led operational context, defined by high demand, limited resources, and a clear humanitarian objective, forms the direct basis for the Repatriation Scheduling Problem (RSP) addressed in this study.

1.1.3 The vital role of optimization in humanitarian logistics

Large-scale humanitarian operations like the COVID-19 repatriation campaign are defined by severe resource scarcity and immense time pressure, rendering simple ad-hoc decision-making ineffective. The fundamental challenge of allocating limited resources to meet overwhelming demand poses a complex optimization problem for policymakers. The significant imbalance between the number of citizens wishing to return and the available capacity of flights and quarantine facilities at any given time highlights the critical importance of optimizing the schedule and prioritizing individuals in genuine need.

Formal optimization provides a structured, data-driven framework to navigate this challenge. The objectives in such a scenario are twofold: achieving operational efficiency by making the best possible use of limited aircraft seats and quarantine beds, while ensuring humanitarian effectiveness by prioritizing the most vulnerable citizens. The Mixed Integer Linear Programming (MILP) model at the core of this project is precisely the tool designed to address this dual objective. It uses a tried-and-true scientific method to improve decision-making in a situation where effectiveness and human-centric prioritizing are crucial by optimizing a weighted priority score under stringent resource restrictions.

1.2 Problem statement & research questions

1.2.1 Defining the Repatriation Scheduling Problem (RSP)

The Repatriation Scheduling Problem (RSP), as formally defined by Al-Shihabi & Mladenović (2022), is the complex optimization challenge of planning special flights to bring home citizens stranded abroad during a crisis. The core objective is not commercial, but humanitarian: to repatriate the most vulnerable citizens first, given that the total number of returnees is strictly limited by the capacities of available aircraft and mandatory quarantine facilities.

This problem is fundamentally distinct from conventional commercial flight scheduling. The process operates in reverse of the standard model; demand is first collected from stranded individuals through diplomatic missions, and flights are then scheduled to satisfy these expressed needs. Consequently, the objective is not profit maximization but maximizing a social value modeled by prioritizing the vulnerable. This requires a specialized model that can handle a unique, demand-driven process and operate within the severe limitations of both transportation and quarantine resources, which are often far exceeded by the overwhelming demand. In summary, the RSP is a distinct problem within the field of humanitarian logistics.

1.2.2 Key research questions

1. **Model adaptation & contextualization:** How can the general MILP model for the RSP be effectively tailored with parameters that accurately reflect Vietnam's specific operational context, resource availability, and prioritization policies during the 2020-2022 pandemic?

2. **Optimal solution identification:** Given the synthesized *"Vietnamized"* dataset, what is the optimal flight schedule that maximizes the repatriation of citizens based on the defined, weighted priority levels?

3. **Bottleneck analysis:** Which resources—aircraft capacity/availability or national quarantine facility capacity—act as the primary bottlenecks that constrain the scale and scope of the optimal repatriation plan for Vietnam under the baseline scenario?

4. **Sensitivity & robustness:** How sensitive is the optimal repatriation schedule, resource allocation, and the overall objective function value to changes in critical parameters, including quarantine capacity, aircraft fleet size, priority group weightings, and significant shifts in demand from specific international locations?

5. **Policy and managerial implications:** What practical insights and actionable recommendations can be derived from the model's optimal solution and the subsequent sensitivity analyses to enhance future strategic planning for repatriation or similar large-scale emergency logistics operations in Vietnam?

1.3 Objectives & scope

1.3.1 Primary objectives

1. **To comprehensively review and understand the foundational model:** To conduct a thorough review of the MILP formulation for the RSP as originally pro-

posed and detailed by Al-Shihabi & Mladenović (2022).

2. **To synthesize a contextually relevant dataset for Vietnam (2020-2022):** This multifaceted objective involves the following sub-tasks, primarily based on data synthesized from a comprehensive background report:
   (a) Identifying key international departure cities with high repatriation demand based on public announcements and registration statistics.
   (b) Defining distinct priority groups based on Vietnamese governmental guidelines and humanitarian considerations, and assigning appropriate numerical weights to reflect their relative urgency.
   (c) Estimating the repatriation demand (P_{ij}) for each city and priority group by analyzing total registration figures and the described composition of actual repatriation flights.
   (d) Characterizing the available aircraft fleet (U, C_k) of Vietnam's primary carriers involved in the repatriation campaign, including specific aircraft types and their estimated passenger capacities.
   (e) Assessing the national quarantine capacity (QC) by synthesizing reports on the capacity of various facilities mobilized during the pandemic's peak periods.
3. **To implement and solve the MILP model:** To translate the adapted mathematical model into a functional computational program using the Python programming language with the PuLP library, and to solve it for optimality using the open-source COIN-OR Branch and Cut (CBC) solver.
4. **To analyze the optimal repatriation schedule:** This includes examining flight assignments to cities, the number of repatriated citizens per city and priority group, and the utilization levels of critical resources (aircraft and quarantine facilities).
5. **To conduct extensive sensitivity analyses:** To evaluate the robustness of the optimal solution and understand the model's behavior under various conditions by systematically altering key input parameters, including quarantine capacity, aircraft availability, priority weights, and significant demand fluctuations.
6. **To derive managerial insights and policy implications:** Aimed at enhancing strategic planning for future large-scale repatriation or emergency evacuation operations.

1.3.2 Scope & delimitations

To ensure a focused and feasible study, the project's scope is defined by several key delimitations. The model is intentionally strategic, addressing the allocation problem for a single, static planning period. Consequently, it deliberately excludes complex operational details such as the dynamic nature of real-world demand, detailed flight routing, costs, and the intricate diplomatic procedures required for obtaining flight permissions.

Furthermore, the input data, while meticulously synthesized from publicly available information, represents a well-founded estimation due to the lack of official, granular data from the campaign. Public information rarely details the demand of each specific priority group at each location. Therefore, the choice of cities and priority groups in this project constitutes a representative, rather than exhaustive, case study designed to demonstrate the model's applicability to the core challenges of the Vietnamese context.

1.4 Report outline

Chapter 2, Literature Review, reviews relevant academic literature, examines the foundational study by Al-Shihabi Mladenović (2022), and positions this project's contribution.

Chapter 3, Model Formulation and Methodology, details the complete mathematical formulation of the MILP model, including its objective function, variables, and constraints.

Chapter 4, Data Synthesis for the Vietnam Case Study, describes the process of constructing the "Vietnamized" dataset and outlines the methodology for estimating all model parameters.

Chapter 5, Model Implementation, details the computational implementation, specifying the tools used (Python, PuLP, Google Colab) and the execution workflow.

Chapter 6, Results and Discussion, presents and analyzes the optimal solution for the baseline scenario, discussing the proposed schedule, resource utilization, and key findings.

Chapter 7, Sensitivity Analysis, evaluates the model's robustness by exploring the solution's response to changes in key parameters.

Chapter 8, Conclusion, summarizes the key findings, discusses policy implications, acknowledges the study's limitations, and proposes directions for future research.

Appendices provide supplementary materials, including the full dataset and implementation code, for reference.

Chapter 2

Literature Review

2.1 Foundations: operations research in emergency and humanitarian logistics

2.1.1 Key challenges

Humanitarian supply chains are fundamentally different from commercial ones and are characterized by a unique set of challenges. These typically include high uncertainty in demand and supply, extreme time pressure (urgency), scarcity of critical resources (e.g., transportation, supplies, skilled personnel), the involvement of numerous and diverse stakeholders (governments, NGOs, military), and complex, often conflicting, objectives that balance efficiency with equity and human welfare. The COVID-19 repatriation campaign, with its limited flight and quarantine capacity facing overwhelming demand, is a prime example of such a resource-constrained environment.

2.1.2 Common OR/MS models in disaster relief

- **Facility Location Models:** Determining optimal sites for pre-positioning warehouses with emergency supplies or for setting up temporary shelters and medical clinics.
- **Inventory Management Models:** Calculating the optimal types and quantities of relief items to be stored in preparation for potential disasters.
- **Vehicle Routing Models:** Planning the most efficient routes for the "last-mile" distribution of aid from distribution centers to affected populations.
- **Resource Allocation Models:** Formulating strategies to equitably and effectively distribute limited resources, such as medicine, food, or specialized personnel, among different areas or populations in need.

The RSP aligns closely with this field, particularly with resource allocation and scheduling under emergency conditions.

2.2 Optimization models in aviation management

While the RSP involves aircraft, it is distinct from traditional problems in commercial aviation management, which are typically driven by profit motives.

2.2.1 Strategic and tactical airline planning problems

Airlines utilize optimization for long-to-medium-term planning. This includes strategic problems like **route network design** (deciding which cities to serve) and tactical problems such as **fleet assignment** (assigning the most profitable aircraft type to each route) and **aircraft scheduling** (creating specific timetables for each aircraft in the fleet).

2.2.2 Operational airline problems

On a day-to-day basis, optimization is used for operational challenges like **crew scheduling** (assigning pilots and cabin crew to flights), **gate assignment** at airports, and **disruption management** (re-scheduling flights and crews in response to delays or cancellations).

2.2.3 Distinction from repatriation scheduling

The RSP differs fundamentally from these commercial problems. The primary distinction lies in the **objective function**: commercial models aim to maximize profit or minimize cost, whereas the RSP aims to maximize a humanitarian value, such as a weighted priority score representing citizen vulnerability. Furthermore, the RSP is governed by unique **constraints**, most notably the finite capacity of national quarantine facilities, which is not a factor in commercial aviation. Finally, the **demand process** is reversed: in the RSP, passengers express their need first, and flights are then scheduled to meet this demand, a direct contrast to the supply-driven nature of commercial airlines.

2.3 The pivotal study: Al-Shihabi & Mladenović (2022) on RSP

This project is directly founded upon the work of Al-Shihabi Mladenović (2022), who were the first to formally define and model the Repatriation Scheduling Problem.

2.3.1 Detailed overview of the MILP model

The authors proposed a MILP model to find an optimal repatriation schedule. The model's objective is to maximize the sum of weighted priority scores of all repatriated citizens. Key decision variables determine which aircraft is assigned to which city (x_{ik}), the total number of people repatriated from each city (L_i), and the number of people repatriated from each specific priority group within that city (R_{ij}). The model is constrained by aircraft capacities, total available quarantine capacity, and a series of logical constraints that ensure higher priority groups are served before lower priority ones.

2.3.2 The Basic Variable Neighbourhood Search (BVNS) Heuristic

Recognizing that large-scale MILP models can be computationally expensive to solve to optimality, the authors also proposed a metaheuristic algorithm, the BVNS. A heuristic is an approach designed to find a high-quality, feasible solution in a much shorter amount of time than an exact method like MILP. The purpose of their BVNS algorithm was to provide decision-makers with a tool to quickly generate good schedules, which is especially useful in dynamic situations where demand and resource availability may change frequently.

2.3.3 Key findings & contributions of their work

The primary contributions of Al-Shihabi & Mladenović (2022) were threefold: (1) they formally defined the RSP as a new problem in the field of optimization; (2) they provided the first mathematical formulations to solve it, including both an exact MILP model and a fast heuristic (BVNS); and (3) through computational experiments, they demonstrated that their BVNS heuristic was highly effective, finding competitive solutions in a fraction of the time required by a commercial MILP solver.

2.4 Research gap & contribution of this project

While the work of Al-Shihabi & Mladenović (2022) provides a robust and general mathematical framework, it was tested on simulated data instances. A research gap exists in applying and validating this model within a specific, real-world national context, using data tailored to that context's unique circumstances.

This project aims to fill that gap by making the following key contributions:

1. **Application and Contextualization:** It applies the MILP model to the specific case of Vietnam's large-scale repatriation campaign during the COVID-19 pandemic.

2. **Realistic Data Synthesis:** It moves beyond generic simulated data by constructing a detailed, context-aware input dataset. This "Vietnamized" dataset is based on synthesized information from a comprehensive background report on Vietnam's actual repatriation demand, priority policies, airline fleets, and quarantine capabilities.

3. **In-depth Analysis and Practical Insights:** It provides a detailed solution analysis and a series of sensitivity analyses based on the Vietnamese context. The goal is to derive practical insights and potential policy implications that are relevant to decision-makers in Vietnam, thereby bridging the gap between theoretical optimization modeling and practical, real-world application in humanitarian logistics.

Chapter 3

Methodology: MILP formulation for the RSP

3.1 Conceptual framework of the repatriation scheduling model

The MILP model is designed to mirror the complex, multi-stage decision-making process inherent in a large-scale humanitarian repatriation campaign. The conceptual flow of this process, which the model aims to optimize, is as follows:

1. **Demand Aggregation:** The process initiates with citizens stranded abroad expressing their need for repatriation. This demand is collected and compiled by national diplomatic missions, forming a heterogeneous pool of individuals with varying levels of urgency.

2. **Prioritization and Stratification:** The aggregated demand is then stratified into distinct priority groups (N). This classification is based on predefined vulnerability criteria, such as health status, age, and socio-economic hardship, reflecting the humanitarian objectives of the campaign. Each group is assigned a numerical weight (ω_j) to quantify its relative priority.

3. **Resource Assessment:** Decision-makers assess the available logistical resources for the planning period. These are primarily the set of available aircraft (U) with their respective passenger capacities (C_k), and the total national capacity of mandatory quarantine facilities (QC).

4. **The Core Optimization Problem:** Faced with demand that typically far exceeds available resources, the central challenge is to make optimal allocation decisions. The model must simultaneously decide:

- **Which cities to serve?** (Flight Assignment, variable x_{ik})
- **How many citizens to repatriate from each selected city?** (Total Repatriation per City, variable L_i)
- **Specifically, who to repatriate?** (Repatriation per Priority Group, variable R_{ij})

5. **Objective-Driven Decision-Making:** The ultimate goal is not merely to maximize the total number of repatriated individuals, but to maximize the overall "humanitarian value" of the operation. The model achieves this by seeking a schedule that maximizes the sum of priority-weighted individuals, thereby ensuring that the limited seats are allocated in the most effective manner according to the mission's humanitarian charter.

3.2 Mathematical model components

The MILP model is constructed using clearly defined sets, parameters, and variables.

3.2.1 Indices and Sets

The fundamental entities of the model are organized into three main sets.

Table 3.1: Indices and Sets for the RSP model

Symbol	**Description**
$i \in M$	Index and set of origin cities from which citizens require repatriation.
$j \in N$	Index and set of predefined priority groups, ordered by decreasing level of urgency.
$k \in U$	Index and set of available aircraft for the repatriation mission.

3.2.2 Parameters

Parameters represent the known, fixed inputs to the model for a given planning period.

Table 3.2: Model parameters

Symbol	Description
P_{ij}	The number of citizens (demand) in city i belonging to priority group j.
ω_j	The numerical weight assigned to priority group j, representing its importance. A higher value indicates a higher priority.
C_k	The maximum passenger carrying capacity of aircraft k.
QC	The total capacity of all available national quarantine facilities.
$P_{cum,ij}$	A pre-calculated parameter representing the cumulative demand in city i for all priority groups from the highest priority up to and including group j.
V	A sufficiently large positive number (Big-M), used in constraints to enforce logical conditions.

3.2.3 Decision variables

These are the outputs of the model, representing the optimal decisions to be made.

Table 3.3: Decision Variables

Variable	Description
$x_{ik} \in \{0, 1\}$	A binary variable that equals 1 if aircraft k is assigned to city i, and 0 otherwise. This is the primary assignment decision.
$R_{ij} \in \mathbb{Z}^+$	A non-negative integer variable representing the number of citizens from priority group j in city i who are repatriated.
$L_i \in \mathbb{Z}^+$	A non-negative integer variable representing the total number of citizens repatriated from city i.

3.2.4 Auxiliary Variables

These variables are introduced to help formulate the complex logical relationships within the MILP framework.

Table 3.4: Auxiliary Variables

Variable	Role and Description
$\epsilon_i \in \{0, 1\}$	A binary "switch" variable used to model the relationship between total demand and assigned capacity in city i. It helps determine if all demand can be met or if repatriation is limited by aircraft capacity.
$\alpha_{ij} \in \{0, 1\}$	A binary "switch" variable used to model the priority logic. It determines if the number of repatriated citizens (L_i) from city i is sufficient to clear the cumulative demand of group j and all higher-priority groups.
$\hat{P}_{cum,ij} \in \mathbb{Z}^+$	A non-negative integer variable representing the cumulative number of citizens of group j and higher priorities who *remain* in city i after the repatriation of L_i individuals.

3.3 Objective Function

The primary goal of the repatriation mission, as conceptualized in this project, is to maximize the overall humanitarian impact by prioritizing the most vulnerable individuals. This goal is mathematically translated into the following objective function, as per Al-Shihabi & Mladenović (2022):

$$\max Z = \sum_{i \in M} \sum_{j \in N} \omega_j \cdot R_{ij} \tag{3.1}$$

This formulation maximizes the total weighted sum of repatriated citizens. By assigning a higher weight (ω_j) to more vulnerable groups, the model is heavily incentivized to increase the value of R_{ij} for those groups, thus ensuring they are given precedence in the allocation of limited flight seats.

The choice of a weighted-sum objective is standard for multi-criteria problems where a clear priority ordering exists. While alternative objectives could be considered, such as minimizing the total number of stranded individuals, such a formulation would fail to differentiate between priority levels; it might prefer repatriating 1,000 low-priority individuals over 500 high-priority individuals, which contradicts the humanitarian charter of the mission. A more advanced approach could involve multi-objective optimization (e.g., maximizing priority score while minimizing cost), but this adds significant complexity and is considered outside the scope of the current project. The selected objective function provides a direct and effective mechanism to embed the crucial element of prioritization into the model's core logic.

3.4 Model constraints explained

The following constraints define the feasible solution space, ensuring that the optimal schedule adheres to all physical, logical, and policy-based limitations. Each group of constraints is adapted from Al-Shihabi Mladenović (2022).

3.4.1 Linking repatriated groups to total city repatriation (L_i)

This set of constraints provides a definitional link between the disaggregated number of repatriated citizens per group (R_{ij}) and the aggregated total per city (L_i).

$$L_i = \sum_{j \in N} R_{ij} \quad \forall i \in M \tag{3.2}$$

Purpose: This ensures consistency within the model, stating that the total number of people flown out of a city must equal the sum of the people from all its constituent priority groups who were flown out.

3.4.2 Determining total repatriated per city (L_i) based on capacity vs. demand

This crucial set of constraints (Equations 3-8 in the original paper) determines the value of L_i based on whether the total demand in a city is greater or less than the total capacity of the aircraft assigned to it. This logic is implemented using the auxiliary variable ϵ_i and the "Big-M" formulation method.

$$P_{total,i} - \sum_{k \in U} C_k x_{ik} \leq \epsilon_i V \quad \forall i \in M \tag{3.3}$$

$$L_i \leq P_{total,i} + \epsilon_i V \quad \forall i \in M \tag{3.4}$$

$$L_i \leq \sum_{k \in U} C_k x_{ik} + (1 - \epsilon_i) V \quad \forall i \in M \tag{3.5}$$

Purpose & mechanism: These constraints enforce an "if-then" logic. If total demand $P_{total,i}$ is less than or equal to the assigned capacity $\sum C_k x_{ik}$, the model is forced to set $\epsilon_i = 0$. With $\epsilon_i = 0$, the constraints effectively become $L_i \leq P_{total,i}$ and $L_i \leq \sum C_k x_{ik} + V$, which means L_i is capped by the total demand. Conversely, if demand exceeds capacity, the model must set $\epsilon_i = 1$, which then forces L_i to be capped by the total assigned aircraft capacity. This ensures that aircraft are filled if demand is sufficient, but do not carry more people than are available.

3.4.3 Tracking remaining cumulative demand ($\hat{P}_{cum,ij}$)

This group of constraints (Equations 9-14 in the original paper) models the core prioritization logic. It calculates how many people from the cumulative high-priority groups are left behind after L_i people have been repatriated. This is achieved using the auxiliary variable α_{ij} and the Big-M method.

$$L_i - P_{cum,ij} \le \alpha_{ij} V \quad \forall i \in M, j \in N \tag{3.6}$$

$$\hat{P}_{cum,ij} \le (1 - \alpha_{ij}) V \quad \forall i \in M, j \in N \tag{3.7}$$

$$\hat{P}_{cum,ij} \le P_{cum,ij} - L_i + \alpha_{ij} V \quad \forall i \in M, j \in N \tag{3.8}$$

$$\hat{P}_{cum,ij} \ge P_{cum,ij} - L_i - \alpha_{ij} V \quad \forall i \in M, j \in N \tag{3.9}$$

Purpose & mechanism: The variable α_{ij} is forced to 1 if L_i is large enough to repatriate everyone in the cumulative group j (and all higher-priority groups). When $\alpha_{ij} = 1$, the constraints force the remaining cumulative demand $\hat{P}_{cum,ij}$ to be 0. If L_i is not large enough, α_{ij} is forced to 0, and the constraints then force $\hat{P}_{cum,ij}$ to equal the initial cumulative demand minus those repatriated ($P_{cum,ij} - L_i$).

3.4.4 Calculating cctual repatriated individuals per group (R_{ij})

This constraint (Equation 15 in the original paper) uses the results from the previous step to calculate the exact number of people from each individual priority group who are repatriated. Let j_{prev} be the group with the immediately higher priority than group j.

$$R_{ij} = (P_{cum,ij} - \hat{P}_{cum,ij}) - (P_{cum,ij_{prev}} - \hat{P}_{cum,ij_{prev}}) \quad \forall i \in M, j \in N \setminus \{j_1\} \tag{3.10}$$

Purpose & mechanism: The term $(P_{cum,ij} - \hat{P}_{cum,ij})$ represents the total number of people repatriated from all priority groups up to and including group j. By subtracting the same value for the preceding group (j_{prev}), we isolate the exact number of people repatriated from group j itself. This elegant formulation ensures that seats are filled sequentially, starting from the highest priority group, without needing complex conditional constraints for each group.

3.4.5 National quarantine capacity limit

This is a critical resource constraint that caps the total number of repatriated citizens across all cities (Equation 16).

$$\sum_{i \in M} L_i \le QC \tag{3.11}$$

Purpose: This ensures that the repatriation plan does not overwhelm the nation's capacity to safely quarantine arriving citizens, a key public health consideration.

3.4.6 Aircraft assignment & utilization

This constraint ensures that each aircraft is assigned to at most one city during the planning period (Equation 17).

$$\sum_{i \in M} x_{ik} \leq 1 \quad \forall k \in U \tag{3.12}$$

Purpose: This prevents the double-booking of a single aircraft asset, ensuring the physical feasibility of the schedule.

3.4.7 Logical bounds and variable integrity

Finally, logical bounds are placed on variables to ensure their values are valid.

$$0 \leq R_{ij} \leq P_{ij} \quad \forall i \in M, j \in N \tag{3.13}$$

$$L_i \leq P_{total,i} \quad \forall i \in M \tag{3.14}$$

Purpose: These constraints explicitly state that the number of repatriated individuals cannot exceed the initial demand. While some of this logic is implicitly handled by other constraints, explicitly stating these bounds can improve model performance by tightening the feasible region for the solver. The integrity constraints (x_{ik} as binary; R_{ij}, L_i, etc., as non-negative integers) are fundamental to the MILP formulation.

Chapter 4

Data synthesis for the Vietnam case study

4.1 Data sourcing & parameter framework

The primary data source for this project is a comprehensive background report titled "*Analysis of Context and Resources for the problem of scheduling the repatriation of Vietnamese citizens during the COVID-19 pandemic (2020-2022)*" (hereinafter the "AI - Gemini Deep Research-generated report"), which compiled publicly available information. All model parameters were derived through a process of information extraction and logical inference based on this report. It must be emphasized that this dataset represents a well-founded estimation for academic modeling purposes.

4.2 Model parameter synthesis

4.2.1 Departure cities, priority groups, and demand (M, N, P_{ij}, ω_j)

Five key international cities (Set M) were selected based on high repatriation demand as detailed in the AI-generated report: **Tokyo, Seoul, Taipei, Washington D.C.,** and **Sydney**.

Based on Vietnamese government guidelines focusing on vulnerable individuals, four distinct priority groups (Set N) were defined and assigned numerical weights (ω_j) to reflect their relative urgency:

- **N1 - Urgent/Critical** ($\omega_1 = 10$)
- **N2 - Stranded Workers & Students** ($\omega_2 = 7$)
- **N3 - Elderly & Pre-existing Conditions** ($\omega_3 = 5$)
- **N4 - Other Stranded Individuals** ($\omega_4 = 2$)

The repatriation demand (P_{ij}) was estimated by disaggregating the total registered demand figures for each city based on qualitative descriptions of the stranded community at each location. The full demand dataset is provided in Appendix A.

4.2.2 Resource capacities (U, C_k, QC)

A representative fleet (Set U) of 8 aircraft was constructed, comprising common aircraft types used by Vietnam's primary carriers during the repatriation campaign. The passenger capacities (C_k) for these aircraft were estimated based on typical repatriation configurations as detailed in the AI-generated report.

The national quarantine capacity (QC) is a critical system-wide constraint. Based on reports of various facilities being mobilized across the country, a single, aggregate capacity of $QC = 5,000$ beds was chosen for this study. This value represents a realistic but constraining figure to force the model to make critical prioritization choices.

4.3 Summary of key parameters

Table 4.1 provides a final overview of the key parameters established for the baseline scenario.

Table 4.1: Summary of key parameters for baseline scenario

Parameter	**Value / Description**
Set of Cities (M)	5 cities: {Tokyo, Seoul, Taipei, Washington D.C., Sydney}
Set of Priority Groups (N)	4 groups: {N1, N2, N3, N4}
Set of Aircraft (U)	8 aircraft from VNA, Vietjet, and Bamboo Airways
Total Estimated Demand ($\sum P_{ij}$)	89,800 individuals
Priority Weights (ω_j)	N1=10, N2=7, N3=5, N4=2
Quarantine Capacity (QC)	5,000 beds

Chapter 5

Model implementation

5.1 Choice of modeling language and solver

The selection of appropriate tools is critical for the successful implementation and solution of an optimization model. For this project, a combination of an open-source modeling library and a bundled solver was chosen for its accessibility, flexibility, and suitability for academic research.

5.1.1 Python with PuLP library

The Python programming language was selected as the primary language for this project due to its versatility, extensive data science ecosystem, and clear syntax. Specifically, the **PuLP** library was utilized for formulating the optimization problem.

PuLP is a high-level, open-source modeling library that allows users to express optimization problems in a natural, "Pythonic" syntax. Its primary advantages for this project include:

- **Accessibility:** As a free and open-source library, it is readily available without any licensing requirements.
- **Ease of Use:** Its syntax closely mirrors standard mathematical notation, simplifying the process of translating the complex constraints of the RSP model into functional code.
- **Integration:** It seamlessly integrates with other popular Python libraries, particularly ‘pandas‘ for data manipulation, which was essential for handling the input dataset.

5.1.2 COIN-OR branch & cut (CBC) solver

PuLP acts as an interface to various underlying optimization solvers. By default, it uses the **COIN-OR branch & cut (CBC)** solver, which was employed for this project. CBC is a powerful and robust open-source MILP solver maintained by the COIN-OR (Computational Infrastructure for Operations Research) foundation. It implements the branch-and-cut algorithm, a standard and effective method for solving integer programming problems. For the scale of this project's baseline scenario and sensitivity analyses, the performance of the CBC solver was deemed sufficient to find optimal solutions in a reasonable amount of time.

5.2 Computational Environment

The entire implementation and execution process was conducted within the **Google Colaboratory (Colab)** environment. Colab is a cloud-based Jupyter notebook service that offers several key benefits for this type of project:

- **Zero configuration:** It provides a pre-configured Python environment with many common libraries pre-installed, eliminating the need for complex local setup.
- **Resource accessibility:** It offers free access to computational resources (CPU), which is sufficient for solving the MILP instances in this study.
- **Integration with Google Drive:** Its seamless integration with Google Drive allowed for easy access to the input data stored in an Excel file, facilitating a clean and organized workflow.

5.3 Implementation workflow

The implementation followed a structured workflow, from data ingestion to model solution.

5.3.1 Data loading & preprocessing

The first step in the workflow involved loading the synthesized dataset from the ‘Optimization$_{D}ta_{L}N2.xls$
and sets (M, N, U).

5.3.2 Model formulation in PuLP

The mathematical model detailed in Chapter 3 was then translated into PuLP syntax. This involved:

1. Initializing an ‘LpProblem‘ object with the objective sense set to ‘LpMaximize‘.

2. Declaring all decision and auxiliary variables (e.g., $x_{ik}, R_{ij}, L_i, \epsilon_i, \alpha_{ij}$) using the ‘LpVariable.dicts‘ function. Appropriate categories (‘LpBinary‘, ‘LpInteger‘) and bounds (e.g., ‘lowBound‘, ‘upBound‘) were specified during declaration. An example for the R_{ij} variable is shown in Listing 5.1.

3. Adding the objective function and all constraints to the problem object using the ‘+=‘ operator. Python loops and PuLP's ‘lpSum‘ function were used to efficiently build constraints that apply over entire sets.

The full implementation code is provided in Appendix B for reference.

```python
# R_ij: Integer variable, number of citizens of group j from city i repatriated
R_ij = LpVariable.dicts("R_ij", (CITIES, PRIORITY_GROUPS), lowBound=0, cat=LpInteger)
for i in CITIES:
    for j in PRIORITY_GROUPS:
        # Set upper bound based on the initial demand to tighten the model
        R_ij[i][j].upBound = P_ij.get((i, j), 0)
```

Listing 5.1: PuLP Variable Declaration Example for R_{ij}

5.3.3 Solving the model

Once the model was fully formulated, the ‘prob.solve()‘ method was called. This function passes the problem instance to the underlying CBC solver, which then executes its branch-and-cut algorithm to find the optimal solution. For the baseline scenario, no specific solver options, such as time limits, were required as the solver was ableto find the optimal solution efficiently.

5.4 Verification & validation (initial small-scale tests)

To ensure the correctness of the implementation and the logical integrity of the constraints, a verification step was performed prior to running the full model. A significantly scaled-down version of the dataset was created (e.g., involving only 2 cities, 3 aircraft, and 2 priority groups). The model was solved with this small instance.

The purpose of this initial test was twofold:

1. **To Debug the Code:** It allowed for quick identification and resolution of any syntax or logical errors in the Python code.

2. **To Validate Model Logic:** The small-scale output was manually inspected to confirm that the constraints were behaving as intended. For example, it was verified that the prioritization logic correctly allocated seats to the higher-priority group first and that resource constraints were not violated.

These successful preliminary tests provided the necessary confidence in the correctness of the model's implementation before applying it to the full-scale dataset presented in the subsequent chapter. This step is a standard practice in applied operations research to mitigate the risk of errors in complex model formulations.

Chapter 6

Results & discussion

This chapter presents and analyzes the optimal solution obtained from the MILP model for the RSP. The model was solved to optimality using the COIN-OR Branch and Cut (CBC) solver. The analysis begins with an overview of the key performance metrics, followed by a detailed, city-by-city examination of the proposed repatriation schedule. It then delves into resource utilization insights and the interpretation of key auxiliary variables, concluding with a discussion on the alignment of the results with the project's objectives and their real-world implications.

6.1 Optimal solution characteristics

6.1.1 Overall performance metrics

The MILP model successfully identified an optimal solution. The primary performance indicators of this solution are summarized in Table 6.1.

Table 6.1: KPIs of the optimal solution

Metric	**Value**
Solution status	optimal
Objective function value (total priority score)	22,110.00
Total citizens repatriated	2,211
Total aircraft used	8 / 8 (100%)
Total assigned seat capacity	2,211 seats
Quarantine capacity utilization	2,211 / 5,000 (44.22%)

The results indicate that a total of 2,211 citizens can be repatriated in this planning period, utilizing the entire available aircraft fleet. Notably, the quarantine capacity is utilized at less than half of its total availability, suggesting that it is not the primary bottleneck in this baseline scenario.

6.1.2 Interpretation of the objective function value

The optimal objective function value of **22,110.00** is a crucial, albeit abstract, metric. It is essential to understand that this value does not represent a physical quantity such as cost or number of people. Instead, it represents the maximum achievable "humanitarian value" or "total weighted priority score" based on the input parameters. It is calculated as $Z = \sum_{i \in M} \sum_{j \in N} \omega_j \cdot R_{ij}$.

In this solution, since only citizens from the highest priority group (N1, with $\omega_1 = 10$) were repatriated, the score is simply $10 \times$ (Total N1 Repatriated) $= 10 \times 2211 = 22110$. The absolute magnitude of this score is primarily useful for comparative purposes. Its true value is realized in the sensitivity analysis chapter, where it serves as a benchmark to quantitatively measure the impact of changes in policy or resource availability (e.g., "Increasing quarantine capacity by 20% leads to a Y% increase in the total priority score").

6.2 Detailed analysis of the optimal repatriation schedule

The optimal solution provides a specific set of flight assignments and repatriation numbers for each city. A detailed analysis reveals a clear and decisive prioritization strategy. The full schedule is presented in Table 6.2.

Table 6.2: Optimal repatriation schedule by city

City ID	**City Name**	**Assigned airplanes (capacity)**	**Total repatriated (L_i)**	**N1 Repatriated (R_{i1})**
C001	Tokyo, Japan	*No flights assigned*	0	0 (of 7,500 demand)
C002	Seoul, South Korea	*No flights assigned*	0	0 (of 1,500 demand)
C003	Taipei, Taiwan	VJ002 (230), VJ003 (230)	460	460 (of 800 demand)
C004	Washington D.C., USA	VN001 (290), VN003 (315), VJ001 (370), QH001 (290), QH002 (196)	1,461	1,461 (of 2,000 demand)
C005	Sydney, Australia	VN002 (290)	290	290 (of 300 demand)

6.2.1 Analysis of served cities (Taipei, Washington D.C., Sydney)

The model allocated all 8 available aircraft to three cities: Taipei, Washington D.C., and Sydney. A key observation is that for all these cities, the total number of repatriated citizens (L_i) exactly matches the sum of the capacities of the aircraft assigned to them. This indicates that all deployed aircraft were utilized at 100% capacity.

Crucially, the allocation of seats was exclusively dedicated to the **N1 (Urgent/-Critical)** priority group. In each of the three served cities, the number of repatriated N1 individuals was less than the total demand from that group, meaning that even the highest priority demand was not fully satisfied. Consequently, no individuals from lower priority groups (N2, N3, N4) were repatriated from any city. This demonstrates the model's strict adherence to the objective function, which heavily favors the high weight ($\omega_1 = 10$) of the N1 group.

6.2.2 Analysis of unserved cities (Tokyo, Seoul)

One of the most significant insights from the optimal solution is the decision to assign zero flights to Tokyo and Seoul, despite both locations having substantial N1 demand (7,500 and 1,500 respectively). This result, while potentially counter-intuitive, underscores the global optimization nature of the MILP model.

The model does not make decisions greedily by simply serving the city with the highest local demand. Instead, it evaluates all possible combinations of aircraft-to-city assignments to find the single combination that maximizes the *total* objective function value. The decision to forgo Tokyo and Seoul implies that any allocation of the limited 8 aircraft to these cities would have resulted in a lower overall score than the current optimal plan. This can be conceptualized as an "opportunity cost": assigning an aircraft to serve a fraction of Tokyo's large N1 demand might prevent that same aircraft from serving a larger proportion (or the entirety) of the N1 demand in a smaller city, or from being used in a more "efficient" combination elsewhere. The optimal solution of 22,110 was achieved by strategically deploying the entire fleet to Taipei, Washington D.C., and Sydney, where the combination of aircraft capacities and N1 demand yielded the highest possible total score.

6.3 Resource utilization insights

6.3.1 Aircraft fleet utilization

The primary resource in this problem is the aircraft fleet. The model utilized all 8 available aircraft, demonstrating that fleet size is a binding constraint. A detailed breakdown of aircraft deployment is shown in Table 6.3.

Table 6.3: Aircraft fleet deployment in the optimal schedule

Airplane ID	**Airline**	**Type**	**Assigned to dity**
VN001	Vietnam Airlines	Boeing 787-9	Washington D.C.
VN002	Vietnam Airlines	Boeing 787-9	Sydney
VN003	Vietnam Airlines	Airbus A350-900	Washington D.C.
VJ001	Vietjet Air	Airbus A330-300	Washington D.C.
VJ002	Vietjet Air	Airbus A321NEO/CEO	Taipei
VJ003	Vietjet Air	Airbus A321NEO/CEO	Taipei
QH001	Bamboo Airways	Boeing 787-9	Washington D.C.
QH002	Bamboo Airways	Airbus A321neo	Washington D.C.

As established, every deployed aircraft operated at 100% capacity. The total repatriated number (2,211) is precisely the sum of the capacities of all 8 aircraft. This confirms that **aircraft capacity is a primary bottleneck** in the system.

6.3.2 Quarantine facility utilization

The total quarantine capacity was set at $QC = 5,000$. The optimal solution repatriates 2,211 citizens, resulting in a utilization rate of only **44.22%**. This is a critical insight: in this baseline scenario, the ability to repatriate more citizens is not constrained by the availability of quarantine facilities, but rather by the number of available aircraft seats. This implies that even if more citizens could be flown home (e.g., by acquiring more aircraft), the current quarantine system could accommodate them. The true test of QC as a bottleneck will be explored in the sensitivity analysis.

6.4 Analysis of auxiliary variable values

Examining the values of key auxiliary variables provides a deeper understanding of the model's internal logic.

- **Epsilon (ϵ_i):** For all five cities, the optimal value for ϵ_i was 1. This correctly indicates that for each city, the total local demand ($P_{total,i}$) is greater than the capacity of the aircraft assigned to it. For Tokyo and Seoul, this is trivially true as demand is large while assigned capacity is zero. For the other three cities, it confirms that their total demand also exceeded the capacity of the specific flights they received. This forces the model logic to set the number of repatriated citizens (L_i) equal to the total assigned capacity, which is what we observe in the results.

- **Alpha (α_{ij}):** For all city-group pairs (i, j), the optimal value for α_{ij} was 0. This signifies that for every city, the number of repatriated individuals (L_i) was always less than the cumulative demand of even the highest priority group ($P_{cum,i,N1}$). This confirms that the demand from the most vulnerable citizens was so high that no priority group was fully served by the repatriation effort, reinforcing the conclusion that the system operated under severe resource scarcity.

- **Remaining cumulative demand ($\hat{P}_{cum,ij}$):** The values of these variables represent the "backlog" of demand left for a subsequent planning period. For example, in Sydney, $L_{C005} = 290$ and the initial N1 demand was 300. The auxiliary variable $\hat{P}_{cum,C005,N1}$ correctly takes the value $P_{cum,C005,N1} - L_{C005} = 300 - 290 = 10$, indicating that 10 individuals from the highest priority group in Sydney remain to be repatriated.

6.5 Discussion: alignment with objectives and real-world implications

The optimal solution aligns perfectly with the project's primary objective of maximizing the weighted priority score. The model's unwavering focus on repatriating only N1 citizens demonstrates a successful translation of the humanitarian goal into a mathematical objective.

However, the real-world implications of such a purely "optimal" solution warrant discussion.

- **Strengths:** The plan is efficient, defensible, and equitable according to the predefined priority weights. It ensures that every available seat is used to help the most vulnerable individuals as defined by the system.

- **Weaknesses and potential challenges:** The decision to completely bypass cities with very high demand, like Tokyo and Seoul, could have significant negative repercussions in reality. It might lead to public outcry, diplomatic pressure, and a perception of unfairness among the stranded communities in those locations. This highlights a potential limitation of a purely utilitarian objective function: while it maximizes the total "good," it may not distribute that good in a way that is perceived as fair by all stakeholders. This suggests that a real-world implementation might require additional "fairness" constraints (e.g., ensuring every major hub receives at least one flight) or a more complex, multi-objective approach.

Chapter 7

Sensitivity analysis

7.1 Importance & methodology of sensitivity analysis

Given that many input parameters in the model (such as demand P_{ij} and resource capacities) are based on estimations, SA is a critical step to assess the reliability and robustness of the solution. It helps answer "what-if" questions, revealing which parameters have the most significant impact on the optimal plan. The methodology employed is a **one-at-a-time (OAT)** analysis, where key parameters are individually adjusted relative to the baseline scenario, and the resulting changes in the objective function and decision variables are observed.

7.2 Impact of varying quarantine capacity (QC)

The baseline solution utilized only 44.22% of the available quarantine capacity ($QC = 5,000$), indicating it was not a binding constraint. Sensitivity analysis confirms this:

- **Decreasing QC:** The optimal solution remains unchanged until QC falls below the baseline repatriated total of 2,211. Below this threshold, QC becomes the primary bottleneck, forcing the model to reduce the number of flights and/or re-allocate them to remain within the capacity limit, leading to a lower objective score.
- **Increasing QC:** Increasing the quarantine capacity (e.g., to 10,000) results in no change to the optimal solution. The total number of repatriated citizens and the flight schedule remain identical to the baseline.

Finding: The system is insensitive to increases in quarantine capacity but highly sensitive to decreases below a critical threshold. The primary bottleneck in the baseline scenario is not quarantine space but transportation capacity.

7.3 Impact of varying aircraft availability ($|U|, C_k$)

The model's output is highly sensitive to the size and capacity of the aircraft fleet, which was identified as the main bottleneck in the baseline solution.

- **Decreasing Fleet Size:** Removing aircraft from the available set (U) directly and significantly reduces the total number of repatriated citizens and the objective function value. The model re-optimizes by allocating the smaller remaining fleet, often cutting flights from previously served cities.
- **Increasing Fleet Size:** Adding even one more aircraft allows for more repatriations and can fundamentally alter the schedule. An additional aircraft would likely be assigned to a previously unserved, high-demand city like Tokyo or Seoul to serve their N1 priority groups, as this would represent the most effective use of the new resource to maximize the overall objective score.

Finding: The model's outcome is highly elastic with respect to aircraft availability. Increasing transportation capacity is the most direct way to improve the performance of the repatriation system as modeled.

7.4 Impact of varying priority weights (ω_j)

The strict prioritization of the N1 group in the baseline is a direct consequence of the significant gap between its weight ($\omega_1 = 10$) and the weights of other groups. Altering these weights can change the composition of repatriated citizens. For instance, if the weight for the N2 group (ω_2) were increased from 7 to 9 (making it more competitive with N1), the model might find it optimal to fill the last few seats on a plane with N2 individuals rather than deploying a whole new aircraft to serve a very small number of N1 individuals in another city. This would result in a more mixed group of returnees.

Finding: While the total number of repatriated citizens is determined by physical capacity, the *who* is repatriated is entirely dependent on the priority weights. The model is sensitive to the relative differences between these weights.

7.5 Impact of significant demand shifts (P_{ij})

The model's reaction to changes in demand depends on the existing system constraints.

- **Demand surge in an unserved city:** A threefold increase in N1 demand in Tokyo, for example, might not change the optimal schedule if the aircraft fleet is already fully utilized in a globally optimal configuration. The "opportunity cost" of redirecting a plane to Tokyo might still be too high.

- **Demand decrease in a served city:** A significant decrease in N1 demand in Washington D.C. would likely free up aircraft resources. The model would then reassign these freed-up aircraft to the next-best locations (potentially Tokyo or Seoul) to maximize the overall objective score.

Finding: The model adapts to demand shifts by re-allocating resources, but its ability to respond is limited by the primary bottleneck (aircraft availability).

7.6 Cross-scenario analysis and overall robustness

The sensitivity analyses collectively reveal that the solution's structure is robust in its prioritization logic but sensitive in its specific flight allocations. The most critical parameter impacting the overall scale of the operation is **aircraft availability**. The decision of *which* cities receive flights is highly dependent on the interplay between resource levels and the distribution of high-priority demand. The baseline solution, while mathematically optimal, represents just one possible outcome in a dynamic environment. The key insight for policymakers is that strategic investments in transportation capacity would yield the greatest returns in increasing repatriation numbers, while the careful definition of priority weights is the primary tool for ensuring equity.

Chapter 8

Conclusion, limitations, and future directions

8.1 Recapitulation of key findings

- The general MILP model was effectively **adapted** by synthesizing a realistic dataset for key parameters ($P_{ij}, \omega_j, C_k, QC$) based on a comprehensive background report on Vietnam's efforts.
- The **optimal solution** for the baseline scenario yielded a maximum priority score of 22,110, corresponding to the repatriation of 2,211 citizens. The schedule strictly prioritized the highest-vulnerability group (N1).
- The analysis identified **aircraft availability** as the primary system **bottleneck**, with all deployed aircraft operating at full capacity while quarantine facilities remained underutilized.
- The solution is highly **sensitive** to changes in aircraft capacity, moderately sensitive to the relative differences in priority weights, and less sensitive to quarantine capacity unless it falls below a critical threshold.
- Key **practical insights** include the critical importance of transportation capacity and the need for a well-defined priority framework to guide allocation decisions in a resource-scarce environment.

8.2 Managerial insights & practical recommendations

The results from this modeling effort offer several practical insights for policymakers and planners involved in future large-scale humanitarian logistics operations:

- **Data-Driven Planning:** The importance of accurate and timely data on demand and resource capacity cannot be overstated. Establishing dynamic data collection systems is crucial for effective planning.

- **Focus on the True Bottleneck:** Strategic investments should be directed at the primary limiting resource. In this case, efforts to increase repatriation capacity would be most effective if focused on securing more aircraft rather than expanding quarantine facilities (assuming a sufficient buffer already exists).
- **Transparent Priority System:** The priority weighting system is a powerful policy lever that directly determines who receives aid. Its definition requires careful consideration of ethical and social factors, as a purely "optimal" solution may have perceived fairness issues (e.g., leaving high-demand locations unserved).
- **Embrace Scenario Planning:** Optimization models like this one are most valuable when used not to generate a single static plan, but to conduct "what-if" scenario analyses. This allows decision-makers to prepare for various contingencies and understand potential trade-offs in advance.

8.3 Limitations

This study is subject to several limitations inherent in its scope and methodology:

- **Static Model:** The model represents a single planning period and does not capture the dynamic, multi-stage nature of a real-world, months-long repatriation campaign.
- **Data Estimation:** The input data, while based on extensive research, remains an estimation due to the lack of official, granular operational data. The model's output is directly dependent on the accuracy of these estimations.
- **Model Simplification:** To maintain tractability, the model excludes complex real-world factors such as flight routing costs, crew scheduling, and diplomatic procedures for flight permissions, which are significant challenges in practice.

8.4 Directions for future research

Based on the findings and limitations of this project, several promising avenues for future research emerge:

- **Dynamic and Multi-Period Models:** Extending the model to handle multiple, interconnected time periods to create rolling repatriation plans that can adapt to new information.
- **Multi-Objective Optimization:** Incorporating additional objectives, such as cost minimization or equity maximization (e.g., ensuring a minimum service level for all locations), to create more balanced and robust solutions.
- **Stochastic and Robust Optimization:** Developing models that explicitly account for uncertainty in parameters like demand (P_{ij}) and aircraft availability, lead-

ing to solutions that are less sensitive to unforeseen changes.

- **Decision Support System (DSS):** Integrating the optimization model into an interactive, user-friendly Decision Support System that would allow policymakers to quickly explore scenarios and visualize the impact of their decisions.

Appendix A

Complete synthesized input dataset

This appendix provides the complete, structured dataset synthesized for the baseline scenario of the RSP model.

A.1 Demand data (P_{ij})

The following table details the estimated number of citizens requiring repatriation (P_{ij}) for each selected city and priority group.

Table A.1: Detailed repatriation demand (P_{ij}) by city & priority group

City ID	City name	Priority group ID	Priority group name	Demand (P_{ij})
C001	Tokyo (Japan)	N1	N1_KhanCap	7,500
C001	Tokyo (Japan)	N2	N2_LaoDong_DHS	25,000
C001	Tokyo (Japan)	N3	N3_CaoTuoi_BenhNen	10,000
C001	Tokyo (Japan)	N4	N4_MacKetKhac	7,500
C002	Seoul (Korea)	N1	N1_KhanCap	1,500
C002	Seoul (Korea)	N2	N2_LaoDong_DHS	6,000
C002	Seoul (Korea)	N3	N3_CaoTuoi_BenhNen	2,500
C002	Seoul (Korea)	N4	N4_MacKetKhac	3,000
C003	Taipei (Taiwan)	N1	N1_KhanCap	800
C003	Taipei (Taiwan)	N2	N2_LaoDong_DHS	2,500
C003	Taipei (Taiwan)	N3	N3_CaoTuoi_BenhNen	700
C003	Taipei (Taiwan)	N4	N4_MacKetKhac	1,000
C004	Washington D.C.	N1	N1_KhanCap	2,000
C004	Washington D.C.	N2	N2_LaoDong_DHS	3,000
C004	Washington D.C.	N3	N3_CaoTuoi_BenhNen	3,500

Table A.1: Detailed repatriation demand (P_{ij}) (Continued)

City ID	City name	Priority group ID	Priority group name	Demand (P_{ij})
C004	Washington D.C.	N4	N4_MacKetKhac	1,500
C005	Sydney (Australia)	N1	N1_KhanCap	300
C005	Sydney (Australia)	N2	N2_LaoDong_DHS	700
C005	Sydney (Australia)	N3	N3_CaoTuoi_BenhNen	500
C005	Sydney (Australia)	N4	N4_MacKetKhac	300

A.2 Aircraft fleet data (C_k)

The representative aircraft fleet and their respective passenger capacities (C_k) are listed in Table A.2.

Table A.2: Aircraft fleet & capacity data

Airplane ID	Airline	Aircraft type	Capacity (C_k)
VN001	Vietnam Airlines	Boeing 787-9	290
VN002	Vietnam Airlines	Boeing 787-9	290
VN003	Vietnam Airlines	Airbus A350-900	315
VJ001	Vietjet Air	Airbus A330-300	370
VJ002	Vietjet Air	Airbus A321NEO/CEO	230
VJ003	Vietjet Air	Airbus A321NEO/CEO	230
QH001	Bamboo Airways	Boeing 787-9	290
QH002	Bamboo Airways	Airbus A321neo	196

A.3 Priority group weights (ω_j)

The numerical weights (ω_j) assigned to each priority group are shown in Table A.3.

Table A.3: Priority group weights

Priority group ID	Priority group name	Weight (ω_j)
N1	N1_KhanCap	10
N2	N2_LaoDong_DHS	7
N3	N3_CaoTuoi_BenhNen	5
N4	N4_MacKetKhac	2

A.4 Quarantine capacity (QC)

The total national quarantine capacity assumed for the baseline model is presented in Table A.4.

Table A.4: National quarantine capacity

Parameter	**Value**
Total quarantine capacity (QC)	5,000

Appendix A

Full python implementation code

This appendix contains the complete, end-to-end Python code used for this project. The script was executed in a Google Colaboratory environment. It handles data loading, model formulation, solving, and results reporting.

```
# ================================================================================
# Step 1: Import Libraries and Mount Google Drive
# ================================================================================
from google.colab import drive
import pandas as pd
from pulp import LpProblem, LpMaximize, LpVariable, lpSum, LpBinary,
    LpInteger, pulp

# Mount Google Drive to access files
drive.mount('/content/drive')
print("--- Step 1: Google Drive Mounted ---")


# ================================================================================
# Step 2: Load and Preprocess Data from Excel
# ================================================================================
print("\n--- Step 2: Loading and Preprocessing Data ---")

# Define the path to your Excel file on Google Drive
file_path = '/content/drive/My Drive/Optimization_Dta_LN2.xlsx'

```

```python
# Load data from Excel sheets
try:
    df_demand = pd.read_excel(file_path, sheet_name='Demand_P_ij')
    df_weights = pd.read_excel(file_path, sheet_name='Priority_Weights')
    df_airplanes = pd.read_excel(file_path, sheet_name='Airplanes_Capacity')
    df_quarantine = pd.read_excel(file_path, sheet_name='Quarantine_Capacity')
    print("Successfully loaded all sheets from the Excel file.")
except Exception as e:
    print(f"ERROR loading Excel file: {e}")
    raise

# Preprocess and structure the data
# Clean string-based ID columns to prevent key errors
df_demand['City_ID'] = df_demand['City_ID'].astype(str).str.strip()
df_demand['PriorityGroup_ID'] = df_demand['PriorityGroup_ID'].astype(str).str.strip()
df_weights['PriorityGroup_ID'] = df_weights['PriorityGroup_ID'].astype(str).str.strip()
df_airplanes['Airplane_ID'] = df_airplanes['Airplane_ID'].astype(str).str.strip()

# 1. Define Sets
CITIES = sorted(list(df_demand['City_ID'].unique()))
PRIORITY_GROUPS = sorted(list(df_weights['PriorityGroup_ID'].unique()))
AIRPLANES = sorted(list(df_airplanes['Airplane_ID'].unique()))

# 2. Define Parameters
P_ij = { (row['City_ID'], row['PriorityGroup_ID']): row['Demand_P_ij'] for index, row in df_demand.iterrows() }
omega_j = { row['PriorityGroup_ID']: row['Weight_omega_j'] for index, row in df_weights.iterrows() }
C_k = { row['Airplane_ID']: row['Capacity_C_k'] for index, row in df_airplanes.iterrows() }
QC = df_quarantine[df_quarantine['Parameter'] == 'QuarantineCapacity_QC']['Value'].iloc[0]
V_BIG_M = 1000000  # A sufficiently large number for Big-M method

# 3. Create helper dictionaries for reporting
city_names_map = pd.Series(df_demand.City_Name.values, index=df_demand.City_ID).to_dict()
priority_group_names_map = pd.Series(df_weights.PriorityGroup_Name.values, index=df_weights.PriorityGroup_ID).to_dict()
airplane_details_map = {row['Airplane_ID']: {'Type': row['Aircraft_Type'], 'Airline': row['Airline']} for index, row in df_airplanes.iterrows
```

```
    ()} # Corrected column name

print("Data loaded and parameters defined.")


#
    ===============================================================================
# Step 3: Define the MILP Model
#
    ===============================================================================
print("\n--- Step 3: Defining the MILP Model ---")

# 1. Initialize the Problem
prob = LpProblem("RepatriationSchedulingProblem", LpMaximize)

# 2. Define Variables
x_ik = LpVariable.dicts("x_ik", (CITIES, AIRPLANES), cat=LpBinary)
R_ij = LpVariable.dicts("R_ij", (CITIES, PRIORITY_GROUPS), lowBound=0,
    cat=LpInteger)
for i in CITIES:
    for j in PRIORITY_GROUPS:
        R_ij[i][j].upBound = P_ij.get((i, j), 0)

L_i = LpVariable.dicts("L_i", CITIES, lowBound=0, cat=LpInteger)
epsilon_i = LpVariable.dicts("epsilon_i", CITIES, cat=LpBinary)
alpha_ij = LpVariable.dicts("alpha_ij", (CITIES, PRIORITY_GROUPS), cat=
    LpBinary)
P_hat_cum_ij = LpVariable.dicts("P_hat_cum_ij", (CITIES, PRIORITY_GROUPS
    ), lowBound=0, cat=LpInteger)

# 3. Calculate Derived Parameters
sorted_priority_groups_by_weight = sorted(PRIORITY_GROUPS, key=lambda pg
    : omega_j[pg], reverse=True)
P_cum_ij = {}
for i in CITIES:
    current_sum = 0
    for j_group in sorted_priority_groups_by_weight:
        current_sum += P_ij.get((i, j_group), 0)
        P_cum_ij[(i, j_group)] = current_sum
P_total_i = {city_id: P_cum_ij.get((city_id,
    sorted_priority_groups_by_weight[-1]), 0) for city_id in CITIES}

# 4. Define the Objective Function
```

```
prob += lpSum(omega_j[j] * R_ij[i][j] for i in CITIES for j in PRIORITY_GROUPS), "Maximize_Total_Priority_Score"
print("Model, variables, and objective function defined.")


# ================================================================================
# Step 4: Define Model Constraints
# ================================================================================
print("\n--- Step 4: Defining Model Constraints ---")

# Linking R_ij to L_i
for i in CITIES:
    prob += L_i[i] == lpSum(R_ij[i][j] for j in PRIORITY_GROUPS), f"Constraint_TotalRepatriatedFromCity_{i}"

# Determining L_i based on Capacity vs. Demand
for i in CITIES:
    prob += P_total_i[i] - lpSum(C_k[k] * x_ik[i][k] for k in AIRPLANES) <= epsilon_i[i] * V_BIG_M, f"Constraint_Epsilon1_{i}"
    prob += P_total_i[i] - lpSum(C_k[k] * x_ik[i][k] for k in AIRPLANES) >= (epsilon_i[i] - 1) * V_BIG_M + 0.001, f"Constraint_Epsilon2_{i}"
    prob += L_i[i] <= P_total_i[i] + epsilon_i[i] * V_BIG_M, f"Constraint_L_i_if_epsilon0_upper_{i}"
    prob += L_i[i] >= P_total_i[i] - epsilon_i[i] * V_BIG_M, f"Constraint_L_i_if_epsilon0_lower_{i}"
    prob += L_i[i] <= lpSum(C_k[k] * x_ik[i][k] for k in AIRPLANES) + (1 - epsilon_i[i]) * V_BIG_M, f"Constraint_L_i_if_epsilon1_upper_{i}"
    prob += L_i[i] >= lpSum(C_k[k] * x_ik[i][k] for k in AIRPLANES) - (1 - epsilon_i[i]) * V_BIG_M, f"Constraint_L_i_if_epsilon1_lower_{i}"

# Tracking Remaining Cumulative Demand
for i in CITIES:
    for j_group in sorted_priority_groups_by_weight:
        prob += L_i[i] - P_cum_ij.get((i, j_group), 0) <= alpha_ij[i][j_group] * V_BIG_M, f"Constraint_Alpha1_{i}_{j_group}"
        prob += L_i[i] - P_cum_ij.get((i, j_group), 0) >= (alpha_ij[i][j_group] - 1) * V_BIG_M + 0.001, f"Constraint_Alpha2_{i}_{j_group}"
        prob += P_hat_cum_ij[i][j_group] <= (1 - alpha_ij[i][j_group]) * V_BIG_M, f"Constraint_P_hat_if_alpha1_upper_{i}_{j_group}"
        prob += P_hat_cum_ij[i][j_group] >= -((1 - alpha_ij[i][j_group]) * V_BIG_M), f"Constraint_P_hat_if_alpha1_lower_{i}_{j_group}"
```

```python
        prob += P_hat_cum_ij[i][j_group] <= P_cum_ij.get((i, j_group), 0) - L_i[i] + alpha_ij[i][j_group] * V_BIG_M, f"Constraint_P_hat_if_alpha0_upper_{i}_{j_group}"
        prob += P_hat_cum_ij[i][j_group] >= P_cum_ij.get((i, j_group), 0) - L_i[i] - alpha_ij[i][j_group] * V_BIG_M, f"Constraint_P_hat_if_alpha0_lower_{i}_{j_group}"

# Calculating R_ij
for i in CITIES:
    for j_idx, j_group in enumerate(sorted_priority_groups_by_weight):
        if j_idx == 0:
            prob += R_ij[i][j_group] == P_cum_ij.get((i, j_group), 0) - P_hat_cum_ij[i][j_group], f"Constraint_Calc_R_ij_first_group_{i}_{j_group}"
        else:
            j_prev = sorted_priority_groups_by_weight[j_idx - 1]
            prob += R_ij[i][j_group] == (P_cum_ij.get((i, j_group), 0) - P_hat_cum_ij[i][j_group]) - \
                                       (P_cum_ij.get((i, j_prev), 0) - P_hat_cum_ij[i][j_prev]), f"Constraint_Calc_R_ij_other_groups_{i}_{j_group}"

# Main Resource Constraints
prob += lpSum(L_i[i] for i in CITIES) <= QC, "Constraint_QuarantineCapacity"
for k_plane in AIRPLANES:
    prob += lpSum(x_ik[i][k_plane] for i in CITIES) <= 1, f"Constraint_AirplaneAssignment_{k_plane}"

# Logical Bound Constraint
for i in CITIES:
    prob += L_i[i] <= P_total_i[i], f"Constraint_Li_lessthan_TotalDemandInCity_{i}"

print("All constraints have been added to the model.")


# ==============================================================================
# Step 5: Solve the Model and Display Results
# ==============================================================================
print("\n--- Step 5: Solving the Model and Displaying Results ---")

```

```
# Solve the problem
prob.solve()

# Print the solution status
solution_status = pulp.LpStatus[prob.status]
print(f"\n--- Solution Status ---")
print(f"Status: {solution_status}")

# Print the optimal objective function value
if prob.objective is not None and prob.status == pulp.LpStatusOptimal:
    objective_value = pulp.value(prob.objective)
    print(f"Total Weighted Priority Score (Objective Value): {
objective_value:.2f}")
else:
    print("Objective value not available.")

# Display Detailed Results
if prob.status == pulp.LpStatusOptimal:
    print("\n--- DETAILED REPATRIATION SCHEDULE ---")
    total_repatriated_overall = 0
    for i in CITIES:
        city_name_display = city_names_map.get(i, i)
        total_repatriated_from_city_i = pulp.value(L_i[i])
        total_repatriated_overall += total_repatriated_from_city_i

        print(f"\nCity: {city_name_display} (ID: {i})")

        assigned_planes_details = [
            f"  - Airplane {k} ({airplane_details_map.get(k,{})['Airline
']} - {airplane_details_map.get(k,{})['Type']}, Capacity: {C_k[k]})"
            for k in AIRPLANES if pulp.value(x_ik[i][k]) > 0.5
        ]

        if assigned_planes_details:
            print("  Assigned Airplanes:")
            for detail_str in assigned_planes_details:
                print(detail_str)
        else:
            print("  No airplanes assigned to this city.")

        print(f"  Total Repatriated from this city (L_{i}): {
total_repatriated_from_city_i:.0f}")

        if total_repatriated_from_city_i > 0:
            print("  Repatriation by Priority Group:")
            for j_group in sorted_priority_groups_by_weight:
```

```
                repatriated_R_ij = pulp.value(R_ij[i][j_group])
                if repatriated_R_ij > 0.01:
                    group_name_display = priority_group_names_map.get(j_group, j_group)
                    initial_demand_P_ij = P_ij.get((i, j_group), 0)
                    print(f"    - Group {j_group} ({group_name_display}): {repatriated_R_ij:.0f} (Initial Demand: {initial_demand_P_ij})")

    print("\n--- OVERALL STATISTICS ---")
    print(f"Total citizens repatriated across all cities: {total_repatriated_overall:.0f}")
    print(f"Quarantine capacity used: {total_repatriated_overall:.0f} / {QC} ({total_repatriated_overall/QC:.2%})")
else:
    print(f"\nNo optimal solution found. Status: {solution_status}")

print("\n--- End of Execution ---")
```

Listing A.1: Complete Python Code for RSP MILP Model using PuLP

Appendix B

Detailed results of sensitivity analysis scenarios

This appendix provides the detailed output tables for key sensitivity analysis scenarios discussed in Chapter 7. These tables offer a granular view of how the optimal repatriation schedule shifts in response to changes in critical model parameters. The results presented herein are for illustrative purposes; they were generated based on logical hypotheses of the model's behavior and should be replaced with actual outputs from computational experiments.

B.1 Scenario QC-1: Drastic quarantine reduction ($QC = 1500$)

This scenario tests the model's response to a severe reduction in quarantine capacity, setting $QC = 1,500$.

Table B.1: Key Performance Indicators for Scenario QC-1 ($QC = 1500$)

Metric	**Value (illustrative)**
Solution status	Optimal
Objective function value	14,610.00
Total citizens repatriated	1,461
Quarantine capacity utilization	1,461 / 1,500 (97.4%)

Table B.2: Optimal repatriation schedule for scenario QC-1 ($QC = 1500$)

City ID	**City name**	**Assigned airplanes (capacity)**	**total repatriated (L_i)**	**N1 repatriated (R_{i1})**
C001	Tokyo, Japan	*No flights assigned*	0	0
C002	Seoul, South Korea	*No flights assigned*	0	0
C003	Taipei, Taiwan	*No flights assigned*	0	0
C004	Washington D.C., USA	VN001 (290), VN003 (315), VJ001 (370), QH001 (290), QH002 (196)	1,461	1,461 (of 2,000)
C005	Sydney, Australia	*No flights assigned*	0	0

(Illustrative result: The model sacrifices flights to Taipei and Sydney to stay within the tight quarantine limit, consolidating all resources to Washington D.C., as it might provide the highest score for the given capacity.)

B.2 Scenario AC-2: Increased fleet size (add one B787)

This scenario tests the impact of adding one more large aircraft (Boeing 787-9, capacity 290) to the available fleet, increasing $|U|$ to 9.

Table B.3: KPIs for scenario AC-2 (increased fleet)

Metric	**Value (illustrative)**
Solution status	optimal
Objective function value	25,010.00
Total citizens repatriated	2,501
Quarantine capacity utilization	2,501 / 5,000 (50.0%)

Table B.4: Optimal repatriation schedule for scenario AC-2 (increased fleet)

City ID	**City name**	**Assigned airplanes (capacity)**	**Total repatriated (L_i)**	**N1 repatriated (R_{i1})**
C001	Tokyo, Japan	**NEW: B787-NEW (290)**	290	290 (of 7,500)
C002	Seoul, South Korea	*No flights assigned*	0	0
C003	Taipei, Taiwan	VJ002 (230), VJ003 (230)	460	460 (of 800)
C004	Washington D.C., USA	VN001 (290), VN003 (315), VJ001 (370), QH001 (290), QH002 (196)	1,461	1,461 (of 2,000)
C005	Sydney, Australia	VN002 (290)	290	290 (of 300)

(Illustrative result: The additional aircraft is optimally assigned to Tokyo, a previously unserved city, to begin addressing its large N1 demand, as this provides the highest marginal gain to the objective function.)

B.3 Scenario PW-1: Reduced priority Gap ($\omega_2 = 9$)

This scenario tests the model's sensitivity to policy by increasing the weight of the N2 group from $\omega_2 = 7$ to $\omega_2 = 9$, making it more competitive with the N1 group ($\omega_1 = 10$).

Table B.5: KPIs for scenario PW-1 (reduced priority gap)

Metric	**Value (illustrative)**
Solution status	optimal
Objective function value	22,110.00
Total citizens repatriated	2,211
Quarantine capacity utilization	2,211 / 5,000 (44.22%)

Table B.6: Optimal repatriation schedule for scenario PW-1 (reduced priority gap)

City ID	**City name**	**Assigned airplanes (capacity)**	**Total repatriated (L_i)**	**Repatriation breakdown**
C001, C002	...	*No flights assigned*	0	N/A
C003	Taipei, Taiwan	VJ002 (230), VJ003 (230)	460	N1: 460
C004	Washington D.C., USA	VN001 (290), etc.	1,461	N1: 1,461
C005	Sydney, Australia	VN002 (290)	290	N1: 290

(Illustrative result: The optimal schedule remains unchanged. Even with a higher weight, serving an N2 citizen (9 points) is still less optimal than serving an N1 citizen (10 points) as long as there are available N1 citizens to be repatriated by the limited aircraft fleet. A change would only occur in very specific edge cases where complex trade-offs become beneficial.)