Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Optimization Models for Medical Procedures Relocation

Linh Anh Nguyen\textsuperscript{a,b,*}, Andrzej Szalas\textsuperscript{a,c}

\textsuperscript{a}Institute of Informatics, University of Warsaw, 02-097 Warsaw, Poland
\textsuperscript{b}Faculty of Information Technology, Nguyen Tat Thanh University, Ho Chi Minh City, Vietnam
\textsuperscript{c}Department of Computer and Information Science, University of Linköping, SE-581 83 Linköping, Sweden

Abstract

As a side-effect of the Covid-19 pandemic, significant decreases in medical procedures for noncommunicable diseases have been observed. This calls for a decision support assisting in the analysis of opportunities to relocate procedures among hospitals in an efficient or, preferably, optimal manner. In the current paper we formulate corresponding decision problems and develop linear (mixed integer) programming models for them. Since solving mixed integer programming problems is NP-complete, we verify experimentally their usefulness using real-world data about urological procedures. We show that even for large models, with millions of variables, the problems’ instances are solved in perfectly acceptable time.

© 2021 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of KES International.

Keywords: Integer linear programming, Medical information systems, Optimal relocation, Public healthcare

1. Introduction and Motivations

As reported in many sources, there were significant decreases in medical procedures for noncommunicable diseases (Ncd) during the Covid-19 outbreak [3, 6, 10].\footnote{A medical procedure is a course of action intended to improve the health, like a treatment or an operation.} In particular, the World Health Organization report\cite{10} concludes that their study “revealed that three quarters of countries reported a considerable degree of disruption to Ncd services.” According to [3], “hospitals need to be prepared to transfer patients between centers and share resources to optimize the care of regional populations.” It is also emphasized that the situation may be rapidly evolving and that “the severity of the situation and the availability of resources may change on a daily basis.” The review [3] also indicates that “the Covid-19 pandemic has powerfully affected all areas of the healthcare system.” This altogether calls for a decision support assisting in the analysis of opportunities to relocate procedures among hospitals in an efficient or, preferably, optimal manner. One of the challenges for such a support is data integration from heterogeneous data sources about
beds in hospitals, their maximum capacities and completed medical procedures. For experimental purposes, we used data about urological procedures from all Polish public hospitals. However, the models we have developed are largely independent of a particular national healthcare system.

When referring to optimality, one has to specify objective functions to minimize or maximize, where the involved parameter values satisfy given constraints. We indicate and address several types of optimization criteria which may be used in practice. Even though the area of relocating and scheduling has attracted much attention (see, e.g., [8, 9] and references there), we have not found models for particular problems we address. The original contributions of our paper therefore include:

- optimization models for real-world problems originating from the needs to relocate medical procedures for non-communicable diseases among hospitals due to the Covid-19 pandemic outbreak;
- experimental evaluation of the developed models using real-world data.

We use mixed integer programming (MIP) models. As MIP is known to be NP-complete [5, 7], experimental verification of the feasibility of using such models on real-world data is particularly important.

The rest of the paper is structured as follows. First, in Section 2 we provide a high-level formulation of the problems we address and discuss related aspects. Section 3 is dedicated to a detailed problems’ formulation. Next, in Section 4 we develop optimization models for the problems. Section 5 is devoted to a discussion of implementation and experiments. Finally, Section 6 presents final remarks and possible directions for further research and applications.

2. The Addressed Optimization Problems: A High-Level Formulation

The general problems we address can be formulated as follows, where \( \mathcal{R} \) and \( \mathcal{R}' \) are (not necessarily disjoint) sets of geographic regions (e.g., counties), and \( \mathcal{P} \) is a set of medical procedure types specified by the corresponding ICD-9 codes.\(^2\) We abstract from particular objective functions to be optimized, postponing their precise formulations to the discussions of more specific sub-problems.

**Problem 2.1 (High-level formulation).** Assuming that capacities of the medical facilities in \( \mathcal{R} \) are reduced, To what extent the capacities of facilities in \( \mathcal{R}' \) should be increased so that all medical procedures of \( \mathcal{P} \) can be completed in regions \( \mathcal{R} \cup \mathcal{R}' \)? How to relocate the procedures? What relocation is optimal?

**Problem 2.2 (High-level formulation).** Assuming that capacities of the medical facilities in \( \mathcal{R}' \) are increased, To what extent can one reduce the capacities of facilities in \( \mathcal{R} \) so that all medical procedures of \( \mathcal{P} \) can be completed in regions \( \mathcal{R} \cup \mathcal{R}' \)? How to relocate the procedures? What relocation is optimal?

Notice a kind of duality between Problems 2.1 and 2.2. It allows us to focus on variants of Problem 2.1. Models and implementations for analogous variants of Problem 2.2 can be developed as a straightforward adaptation of those for Problem 2.1, presented in this paper.

When an increase or a reduction of the procedures in \( \mathcal{P} \) by \( x \) percent is concerned, possible conflicts in using the same equipment or medical staff should be taken into account. One therefore needs suitable resource consumption measures and assumes that procedures from \( \mathcal{P} \) are limited due to their resource consumption and available capacities. For simplicity, we use one numeric measure \( \text{res}_{\text{cons}}(p) \) to denote the consumption of resources by a medical procedure whose ICD-9 code is \( p \). Thus, an increase (respectively, reduction) of the number of performed procedures from \( \mathcal{P} \) results in the corresponding change of the available resources expressed by \( \sum_{p \in \mathcal{P}} \text{number_ofPerformed}(p) \times \text{res}_{\text{cons}}(p) \), where \( \text{number_ofPerformed}(p) \) denotes the number of actual procedures with ICD-9 code \( p \), performed in a given region during the considered date interval. The measure \( \text{res}_{\text{const}}(p) \) can, for example, be the average number of hospitalization days needed for \( p \) (or a constant in the case when \( p \) usually does not require hospitalization).

Due to the available datasets, in our implementations we use the ICD-9-PL codes.\(^3\) However, the developed models are independent of the choice of ICD-9 variants since it only affects data preprocessing. The list of ICD-9-PL codes...

\(^2\) ICD-9 is the World Health Organization’s Ninth Revision of International Classification of Diseases.

\(^3\) ICD-9-PL is the Polish clinical modification of the WHO ICD-9. It does not affect our models: the preprocessing phase can be adapted to the original WHO ICD-9 schema or any of its national variants, like the US ICD-9-CM.
is divided into: sections (e.g., 55: kidney surgery), subsections (e.g., 55.5: total kidney resection (nephrectomy)), categories (e.g., 55.51: excision of the entire kidney), and subcategories (e.g., 55.512: complete unilateral kidney resection). We assume that preprocessing makes sure that the given set \( P \) consists only of ICD-9 codes with the detail level allowing one to estimate measures like \( \text{res}_\text{const}(p) \) and \( \text{delay}_\text{limit}(p) \) for all \( p \in P \).

### 3. Detailed Problems’ Formulation

In general, our models are designed in a flexible manner so that various objective functions can easily be adopted and implemented. To express objective functions considered in the paper, we assume that the following measures are also available or can be computed from data:

- for \( p \in P \), \( \text{cost}(p) \) is the \( \text{(average) cost of performing a medical procedure with the ICD-9 code } p \);
- for \( r \in R \) and \( r' \in R' \), \( \text{dist}(r, r') \) is the distance between the regions \( r \) and \( r' \).

Let us now provide detailed formulations of Problem 2.1.

**Problem 3.1.** For each region \( r \in R \), the resources to perform in \( r \) all procedures with the ICD-9 code in \( P \) during the date interval \([t_b, t_e)\) are reduced by at least \( \text{dec}_r \) percent. Assuming that for each region \( r' \in R' \), the resources to perform in \( r' \) all procedures with the ICD-9 code in \( P \) can be increased by at most \( \text{inc}_{r'} \) percent starting from the date \( t_b' \), the questions are:

- To what date \( t'_e \) should the number of procedures of \( P \) be increased in \( R' \) in order to relocate to \( R' \) procedures of \( P \) unperformed in \( R \)?
- What is the nearest date \( t'_e \) allowing for a successful relocation?

As preconditions for Problem 3.1, we require that for \( r \in R \), \( t_b < t_e \), and for \( r \in R \cap R' \), \( t_e \leq t_b' \). As a simple use case for Problem 3.1, one may consider dates \( t_1 \), \( t_2 \), \( t'_1 \), and \( t'_2 \) and positive constants \( a \), \( a' \) such that for all \( r \in R \), \( t_b = t_1 \), \( t_e = t_2 \), and for all \( r' \in R' \), \( \text{dec}_r = a \), \( t_b' = t'_1 \) and \( \text{inc}_{r'} = a' \).

The aim of Problem 3.1 is to estimate the nearest date when the increased activities of hospitals in \( R' \) can be reverted to their normal levels. Another closely related problem is to fix the date intervals and look for a common percentage the procedures of \( P \) that should be increased in some regions from \( R' \). It is formally stated below.

**Problem 3.2.** For each region \( r \in R \), the resources to perform in \( r \) all procedures with the ICD-9 code in \( P \) during the date interval \([t_b, t_e)\) are reduced by at least \( \text{dec}_r \) percent. Let \((R'_1, R'_2)\) be a partition of \( R' \). Assuming that for each region \( r' \in R'_1 \), we can increase the resources to perform in \( r' \) all procedures with the ICD-9 code in \( P \) during the date interval \([t_b', t_e')\) by at most \( \text{inc}_{r'} \) percent, the questions are:

- By what percentage \( \text{inc}_2 \) should the number of procedures of \( P \) be increased in each region \( r' \in R'_2 \) in order to relocate to \( R' \) procedures of \( P \) unperformed in \( R' \)?
- What is the smallest value for \( \text{inc}_2 \) allowing for a successful relocation?

As preconditions for Problem 3.2, we require that for \( r \in R \), \( t_b < t_e \), \( t_e \leq \max_{r \in R} t_e' \), and for \( r' \in R' \), \( t_b' < t_e' \), and for \( r \in R \cap R' \), \( t_e \leq t_b' \). As a simple use case for Problem 3.2, one may set \( R'_1 = \emptyset \), \( R'_2 = R' \), dates \( t_1 \), \( t_2 \), \( t'_1 \), \( t'_2 \), and a constant \( a > 0 \) such that \( t_1 < t_2 \), \( t'_1 < t'_2 \), \( t_1 \leq t'_1 \), \( t_2 \leq t'_2 \), for all \( r \in R \), \( t_b = t_1 \), \( t_e = t_2 \) and \( \text{dec}_r = a \) and for all \( r' \in R' \), \( t_b' = t'_1 \), \( t_e' = t'_2 \).

The answers to Problems 3.1 and 3.2 for various parameters give us indicative information about how the number of procedures with the ICD-9 code in \( P \) should be increased in \( R' \) in order to relocate unperformed procedures of \( P \) in \( R \) to \( R' \). Using this information, one can look for the optimal relocation from the point of view of time needed (in Problem 3.1) or the volume increase (in Problem 3.2).

Of course, one may be interested in other objective functions, including:

- the total cost of performing the relocated procedures;
- the total delay time of performing the procedures;
- the total distance of transport needed for the relocation;

\(^4\) Note that some of the objective functions require additional constraints, as specified in the list.
(d) the weighted sum of functions (a)–(c) using given weights \( w_1, w_2 \) and \( w_3 \);
(e) the total cost of performing the relocated procedures assuming additional specific constraints:
   - the total delay time of performing the procedures does not exceed \( \text{time\_limit} \);
   - the total distance of transport needed for the relocation does not exceed \( \text{transport\_limit} \);
(f) the total delay time of performing the procedures assuming additional specific constraints:
   - the total cost of performing the relocated procedures does not exceed \( \text{cost\_limit} \);
   - the total distance of transport needed for the relocation does not exceed \( \text{transport\_limit} \);
(g) the total distance of transport needed for the relocation assuming additional specific constraints:
   - the total cost of performing the relocated procedures does not exceed \( \text{cost\_limit} \);
   - the total delay time of performing the procedures does not exceed \( \text{time\_limit} \).

In the next problem we illustrate the use of these functions.

**Problem 3.3.** For each region \( r \in \mathcal{R} \), the resources to perform in \( r \) all procedures with the ICD-9 code in \( \mathcal{P} \) during the date interval \([t_b, t_e]\) are reduced by at least \( \text{dec}\% \) percent. Assuming that for each region \( r' \in \mathcal{R}' \), the resources to perform in \( r' \) all procedures with the ICD-9 code in \( \mathcal{P} \) during the date interval \([t'_b, t'_e]\) can be increased by at most \( \text{inc}\% \) percent, the questions are:

- Is it possible to relocate unperformed procedures of \( \mathcal{P} \) in \( \mathcal{R} \) to \( \mathcal{R}' \)?
- What is the optimal relocation (w.r.t. minimizing objective functions (a)–(g) listed above)? \( \square \)

Preconditions for Problem 3.3 are the same as for Problem 3.2. In a typical scenario, the user chooses values for the additional parameters (i.e., \( w_1, w_2, w_3, \text{cost\_limit}, \text{time\_limit}, \text{transport\_limit} \)) for the subproblems (d)–(g) after taking into account the answers for the subproblems (a)–(c). Notice that optimal values computed for (a)–(c) may differ in orders of magnitude. Therefore, when choosing weights \( w_1, w_2 \) and \( w_3 \) for the subproblem (d), one should consider the answers for the subproblems (a)-(c) after a suitable normalization.

4. The Models

In this section, we show how to formulate Problems 3.1–3.3 stated in the previous section as (mixed integer) linear programming models, possibly with an outer loop for iterative deepening search.

Let \( \text{period\_len} \) be a parameter for representing the number of days in one time period, e.g. 30 or 7. Time periods are successive, starting from a certain date.\(^5\) We assume that using the past data (from the non-epidemic time period) one can estimate the following forecasts:

- \( \text{forecast}(p, r, t_1, t_2) \), for \( p \in \mathcal{P}, r \in \mathcal{R} \cup \mathcal{R}' \) and a date interval \([t_1, t_2]\), as a forecast of the number of performed procedures of \( p \) in the region \( r \) in the date interval \([t_1, t_2]\);
- \( \text{upper\_forecast}(p, r) \), for \( p \in \mathcal{P} \) and \( r \in \mathcal{R} \cup \mathcal{R}' \), as a forecast of the upper bound of the number of performed procedures of \( p \) in the region \( r \) in one time period.

We also assume that the constraints about \( \text{delay\_limit}(p) \) for \( p \in \mathcal{P} \) only with the accuracy of one time period are to be satisfied and the dates \( t_b, t_e, t'_b, t'_e \) mentioned in Problems 3.1–3.3 are the starting dates of respective time periods.

4.1. Model for Problem 3.1

In order to develop a model for Problem 3.1, let us first consider two cases where this problem can be formulated as a \( \text{MIP} \) model. The first case allows us to use fewer variables and can be solved more efficiently. Then we consider the general case where we use an outer loop for iterative deepening search using a \( \text{MIP} \) model.

In the first case we assume that:

- (i) the constraints about \( \text{delay\_limit}(p) \) for \( p \in \mathcal{P} \) are not important;

\(^5\) These settings facilitate our presentation. In our implementation, however, a time period is either a month or a week and we use operations on time periods instead of dates.
(ii) upper_forecast(p, r) abbreviates forecast(p, r, t_a, t_b) for any r ∈ R_r and any time period [t_a, t_b] (of the same length period_len);

(iii) there are dates t_1, t_2 and t'_1 such that: t_1 ≤ t'_1; t_b_r = t_1 and t_e_r = t_2 for all r ∈ R_r; and t'_b_r = t'_1 for all r' ∈ R_r'.

We will refer to Problem 3.1 of this case as Problem 3.1(a). We formulate the corresponding MIP model as follows:

- **Variables:** for p ∈ P, r ∈ R and r' ∈ R_r, the variable move_{p,r,r'} represents the number of procedures of type p to be relocated from r to r'; in addition, the variable te_r represents the final date of the time interval for performing the relocated procedures.

- **Constraints:**
  - for each p ∈ P, r ∈ R and r' ∈ R_r: move_{p,r,r'} ≥ 0;
  - for each r ∈ R: \( \sum_{r' \in R_r} \sum_{p \in P} move_{p,r,r'} \cdot res_cons(p) \geq \sum_{p \in P} forecast(p, r, t_1, t_2) \cdot res_cons(p) \cdot dec_r / 100; \)
  - for each r' ∈ R_r:
    \( \sum_{p \in P} \sum_{r' \in R_r} move_{p,r,r'} \cdot res_cons(p) \leq \sum_{p \in P} \left( upper_forecast(p, r') / period_len \right) \cdot (te_r - t'_1) \cdot res_cons(p) \cdot inc_r / 100; \)
  - additionally: t'_1 < te_r and t_2 ≤ te_r.

- **Objective function to be minimized:** te_r.

Now consider the second case, in which we still assume the conditions (i)–(ii) about delay_limit and upper_forecast, as for the first case, and instead of the condition about the existence of t_1, t_2 and t'_1 we assume that max t'_b_r ≤ max t_e_r.

Let \( tb_{min} \equiv \min_{r \in R} t_b_r; \) te_max \equiv max_{r \in R} te_r; k \equiv \frac{te_{max} - tb_{min}}{period_len}. \) By the assumptions, t'_b_r ≤ te_max for all r' ∈ R_r'.

We will refer to Problem 3.1 of the considered case as Problem 3.1(b). We formulate it as a MIP model as follows, where successive time periods are numbered from 0 w.r.t. the start date tb_{min}. For a and b being integers such that a ≤ b, the notation a .. b stands for the set \{a, a + 1, ..., b\}.

- **Variables:**
  - for p ∈ P, r ∈ R, r' ∈ R_r and integers i, i' ∈ [0, k), move_{p,r,i,r',i'} represents the number of procedures of p in the region r in the i-th time period to be relocated to the region r' in the i'-th time period;
  - for p ∈ P, r ∈ R, r' ∈ R_r and i, i' ∈ 0..(k - 1) \cup \{+\}: move_{p,r,i,r',i'} ≥ 0;
  - for p ∈ P, r ∈ R, r' ∈ R_r and i, i' ∈ 0..(k - 1) \cup \{+\}: move_{p,r,i,r',i'} = 0 if \( \text{tb}_{min} + i \cdot \text{period_len} < \text{tb}_{r}, \) or \( \text{tb}_{min} + i \cdot \text{period_len} \geq \text{te}_r, \) or \( \text{tb}_{min} + i \cdot \text{period_len} < \text{tb}_{r}', \) or \( i' < i; \)
  - for p ∈ P, r ∈ R, r' ∈ R_r and i ∈ 0..(k - 1) \cup \{+\}: move_{p,r,i,r',i'} = 0 if \( \text{tb}_{min} + i \cdot \text{period_len} < \text{tb}_{r}, \) or \( \text{tb}_{min} + i \cdot \text{period_len} \geq \text{te}_r, \)
  - for r ∈ R and i ∈ \( \left( \frac{\text{tb}_{r} - \text{tb}_{min}}{\text{period_len}} \right), \frac{\text{tb}_{min} + (i + 1) \cdot \text{period_len}}{\text{period_len}} \) and \( I \equiv 0..(k - 1) \cup \{+\}:
    \sum_{p \in P} \sum_{r' \in R_r} \sum_{i \in I} move_{p,r,i,r',i'} \cdot res_cons(p) \geq \frac{\sum_{p \in P} \sum_{r' \in R_r} \sum_{i \in I} \left[ forecast(p, r, \text{tb}_{min} + i \cdot \text{period_len}, \text{tb}_{min} + (i + 1) \cdot \text{period_len}) \cdot res_cons(p) \cdot dec_r / 100; \right];}{6}
  - for r' ∈ R_r and i' ∈ \( \left( \frac{\text{tb}_{r}' - \text{tb}_{min}}{\text{period_len}} \right), \frac{\text{tb}_{min} + (i' + 1) \cdot \text{period_len}}{\text{period_len}} \) \cdot res_cons(p) \cdot inc_r / 100; \)

\(^6\) The interval \{te_{max}, te_r\} may consist of more than one time period. Therefore we use the symbol + rather than a numeric index.
where successive time periods are numbered from 0 w.r.t. the start date \( t_{\text{b min}} \leq t_{\text{e max}} \),

- Objective function to be minimized: \( t_{\text{e}} \)

Variables: \( a \), the notation \( b 

- \text{Variables: for } p \in \mathcal{P}, r \in \mathcal{R}, r' \in \mathcal{R}' \) and integers \( i, i' \in [0, k) \), the variable \( \text{move}_{p, r, i, r', i'} \) represents the number of procedures of type \( p \) in the region \( r \) in the \( i \)-th time period to be relocated to the region \( r' \) in the \( i' \)-th time period.

- Constraints:
  - for \( p \in \mathcal{P}, r \in \mathcal{R}, r' \in \mathcal{R}' \) and integers \( i, i' \in [0, k) \):
    - if \( t_{\text{min}} + i \cdot \text{period}_\text{len} < t_{\text{b}}, \text{ or } t_{\text{min}} + i \cdot \text{period}_\text{len} \geq t_{\text{e}}, \text{ or } t_{\text{min}} + i \cdot \text{period}_\text{len} < t_{\text{b}}' \) or
      - \( (i' - i) \cdot \text{period}_\text{len} > \text{delay}_\text{limit}(p) \) or \( i' < i \),
    - then include the constraint \( \text{move}_{p, r, i, r', i'} = 0 \), else include the constraint \( \text{move}_{p, r, i, r', i'} \geq 0 \);
  - for \( r \in \mathcal{R} \) and \( i \in \left( \frac{t_{\text{b}} - t_{\text{min}}}{\text{period}_\text{len}} \right) \cdot \left( \frac{t_{\text{e}} - t_{\text{min}}}{\text{period}_\text{len}} - 1 \right) \):
    - \( \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} \sum_{i' \in [0, k-1]} \text{move}_{p, r, i, r', i'} \cdot \text{res}_\text{cons}(p) \geq \)
    - \( \sum_{p \in \mathcal{P}} \left[ \text{forecast}(p, r, t_{\text{min}} + i \cdot \text{period}_\text{len}, t_{\text{min}} + (i+1) \cdot \text{period}_\text{len}) \cdot \text{res}_\text{cons}(p) \cdot \text{dec}_r/100 \right] \);
  - for \( r' \in \mathcal{R}' \) and \( i' \in \left( \frac{t_{\text{b}}' - t_{\text{min}}}{\text{period}_\text{len}} \right) \cdot (k - 1) \):
    - \( \sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}} \sum_{i' \in [0, k-1]} \text{move}_{p, r, i, r', i'} \cdot \text{res}_\text{cons}(p) \leq \)
    - \( \sum_{p \in \mathcal{P}} \left[ \text{forecast}(p, r', t_{\text{min}} + i' \cdot \text{period}_\text{len}, t_{\text{min}} + (i'+1) \cdot \text{period}_\text{len}) \cdot \text{res}_\text{cons}(p) \cdot \text{inc}_r/100 \right] \).

We now deal with Problem 3.1 for the case when the first two Min formulations given in this subsection are not applicable. We will refer to Problem 3.1 of this general case as Problem 3.1(c). Our method for solving the problem uses iterative deepening search and is formulated as follows:

- if the problem \( P_{3.1}(te') \) is infeasible for a big enough \( te' \) which is the beginning of a time period, then return that Problem 3.1 is infeasible for the given inputs;
- initialize \( te' \) to the earliest starting date of a time period not sooner than \( te'_0 \), where \( te'_0 \) is the date defined by (1);
- repeat (note that the feasibility check guarantees the termination of this loop):
  - if the problem \( P_1(te') \) is feasible then return \( te' \);
  - \( te' := te' + \text{period}_\text{len} \).

If this process returns a date (i.e., not “infeasibility”), then the result may be suboptimal. However, the returned date is later than the optimal date and differs from the optimal value by less than one time period.

### 4.2. Model for Problem 3.2

In order to develop a model for Problem 3.2 let us introduce the following notation:
We formulate Problem 3.2 as a MIP problem as follows, where successive time periods are numbered from 0 w.r.t. the starting date $t_{\text{min}}$.

- **Variables:** for $p \in P$, $r \in R$, $r' \in R'$ and integers $i, i' \in \{0, k\}$, the variable $move_{p,r,i,r',i'}$ represents the number of procedures of $p$ in the region $r$ in the $i$-th time period to be relocated to the region $r'$ in the $i'$-th time period; in addition, $inc_2$ represents a percentage of the increase of procedures.

- **Constraints:**
  
  (i) for $p \in P$, $r \in R$, $r' \in R'$ and integers $i, i' \in \{0, k\}$:
  
  $\sum_{p \in P} \sum_{r \in R} \sum_{i \in \{0, (k-1)\}} move_{p,r,i,r',i'} \cdot res_{\text{cons}}(p) \geq \sum_{p \in P} [\text{forecast}(p, r, t_{\text{min}} + i \cdot period_{\text{len}}, t_{\text{min}} + (i+1) \cdot period_{\text{len}}) \cdot res_{\text{cons}}(p) \cdot \text{dec}_r / 100];$

  (ii) for $r \in R$ and $i \in (t_{\text{min}} \cdot period_{\text{len}} - 1) / (t_{\text{min}} \cdot period_{\text{len}})
  \sum_{p \in P} \sum_{r \in R} \sum_{i \in \{0, (k-1)\}} move_{p,r,i,r',i'} \cdot res_{\text{cons}}(p) \leq \sum_{p \in P} [\text{forecast}(p, r', t_{\text{min}} + i' \cdot period_{\text{len}}, t_{\text{min}} + (i'+1) \cdot period_{\text{len}}) \cdot res_{\text{cons}}(p) \cdot \text{inc}_r / 100];$

  (iii) for $r' \in R'_1$ and $i' \in (t_{\text{min}} \cdot period_{\text{len}} - 1) / (t_{\text{min}} \cdot period_{\text{len}})
  \sum_{p \in P} \sum_{r \in R} \sum_{i \in \{0, (k-1)\}} move_{p,r,i,r',i'} \cdot res_{\text{cons}}(p) \leq \sum_{p \in P} [\text{forecast}(p, r', t_{\text{min}} + i' \cdot period_{\text{len}}, t_{\text{min}} + (i'+1) \cdot period_{\text{len}}) \cdot res_{\text{cons}}(p) \cdot \text{inc}_2 / 100].$

- **Objective function to be minimized:** $inc_2$.

### 4.3. Model for Problem 3.3

Let $t_{\text{min}}, t_{\text{max}}$ and $k$ be specified as in Section 4.2. We formulate Problem 3.3 as a MIP problem as follows.

- **Variables:** $move_{p,r,i,r',i'}$ as specified in Section 4.2.

- **Constraints:**
  
  - the constraints specified in the items (i) and (ii) listed in Section 4.2;
  - the constraints specified in the item (iii) listed in Section 4.2 for a larger range $r' \in R'$ (instead of $r' \in R'_1$);
  - if the considered subproblem is (e) or (g) (see page 4), then include:
    
    $\sum_{p \in P} \sum_{r \in R} \sum_{i,j \in \{0, (k-1)\}} move_{p,r,i,r',i'} \cdot (i' - i) \cdot \text{period}_{\text{len}} \leq \text{time}_{\text{limit}};$

  - if the considered subproblem is (e) or (f), then include:
    
    $\sum_{p \in P} \sum_{r \in R} \sum_{i,i' \in \{0, (k-1)\}} move_{p,r,i,r',i'} \cdot \text{dist}(r, r') \leq \text{transport}_{\text{limit}};$

  - if the considered subproblem is (f) or (g), then include:
\[
\sum_{p \in P} \sum_{r \in R} \sum_{r' \in R'} \sum_{i,i' \in 0..(k-1)} move_{p,r,i,i'} \cdot \text{cost}(p) \leq \text{cost\_limit}.
\]

- **Objective function to be minimized:**

1. if the considered subproblem is (a) or (e): \[\sum_{p \in P} \sum_{r \in R} \sum_{r' \in R'} \sum_{i,i' \in 0..(k-1)} move_{p,r,i,i'} \cdot \text{cost}(p);\]

2. if the considered subproblem is (b) or (f): \[\sum_{p \in P} \sum_{r \in R} \sum_{r' \in R'} \sum_{i,i' \in 0..(k-1)} [move_{p,r,i,i'} \cdot (i' - i) \cdot \text{period\_len}];\]

3. if the considered subproblem is (c) or (g): \[\sum_{p \in P} \sum_{r \in R} \sum_{r' \in R'} \sum_{i,i' \in 0..(k-1)} move_{p,r,i,i'} \cdot \text{dist}(r,r');\]

4. if the considered subproblem is (d): \[w_1 \cdot f_1 + w_2 \cdot f_2 + w_3 \cdot f_3,\] where \(f_1, f_2\) and \(f_3\) are the expressions specified in the above items 1, 2 and 3, respectively.

5. **Experimental Results**

We have implemented our models in **Python** using **PostgreSQL** as a database management system, with the following assumptions and settings:

- \(\text{res\_cons}(p), \text{delay\_limit}(p), \text{cost}(p)\) and \(\text{dist}(r,r')\), for \(p \in P, r \in R\) and \(r' \in R'\), can be read or estimated from the underlying database;

- \(\text{forecast}(p, r, t_1, t_2)\), for \(p \in P, r \in R \cup R'\) and each time period \([t_1, t_2)\) in the considered date interval, has been computed or predicted by another module/model and stored in the database.

As the MIP solver we use **Cplex** [2] via the PuLP package.  

We have conducted performance tests of our implementation using the following settings:

- \(R\) and \(R'\) are sets of counties in Poland;
- \(P\) is a set of ICD-9-PL categories (or subsections that do not have categories) of urology;
- \(\text{forecast}(p, r, t_1, t_2)\), for \(p \in P, r \in R \cup R'\) and each time period \([t_1, t_2)\) in the considered date interval, has been estimated using the real data from the years 2009-2018;
- due to incomplete data,
  - either \(\text{res\_cons}(p)\) and \(\text{cost}(p)\) are set to 1 and \(\text{delay\_limit}(p)\) is set to 5 years for all \(p \in P\);
  - or \(\text{res\_cons}(p)\) and \(\text{cost}(p)\) are set to random integers in the range \([0,100)\) and \(\text{delay\_limit}(p)\) is set to a random number of months in the range \([0,12)\) for all \(p \in P\);
- the time period is one month;
- the MIP relative optimality gap of Cplex is set to 0.05;
- the tests are done on the IBM Power System AC922 (model: 8335-GTH) with two 16-core 2.7 GHz (3.3 Turbo) POWER9 Processors, equipped with 1TB RAM.

In experiments we use the following sets and settings:

- \(P_1\) denotes the set of all ICD-9-PL categories (or subsections that do not have categories) of the ICD-9-PL section 55 (kidney surgery), where \(\text{res\_cons}(p)\) and \(\text{cost}(p)\) are set to 1 and \(\text{delay\_limit}(p)\) is set to 5 years for all \(p \in P_1\);
- \(P_2\) is the same as \(P_1\) but with \(\text{res\_cons}(p)\) and \(\text{cost}(p)\) set to random integers in the range \([0,100)\), and \(\text{delay\_limit}(p)\) is set to a random number of months in the range \([0,12)\) for all \(p \in P_2\);
- \(P_3\) (respectively, \(P_4\)) is the extension of \(P_1\) (respectively, \(P_2\)) that additionally covers the ICD-9-PL sections 56 (ureter surgery) and 57 (bladder surgeries and procedures);
- \(R_1\) denotes the set of all (42) counties of the Mazowieckie province of Poland, setting that \(tb_r = 2020-03-01, te_r = 2020-07-1\) and \(\text{dec}_r = 30\%\) for all \(r \in R_1\).  

---

7 Of course, one can use other solvers, too. We have chosen Cplex due to its known very good performance.

8 We use percentages for better readability. In models we use \(\text{dec}_r/100\), so e.g., 30% is represented as 30 rather than 0.3.
• \( R'_i \) denotes the set of all counties (24) of the Łódź province of Poland, setting that \( t\beta'_1 = 2020-05-01 \) and \( inc'_i = 50\% \) for all \( r \in R'_i \);
• \( R'_2 \) differs from \( R'_1 \) by assuming that \( t\beta'_2 = 2020-06-01 \) for \( r \) being the Łódź county (of the Łódź province);
• \( R'_3 \) denotes the set of all counties of the Łódź province setting that \( t\beta'_3 = 2020-05-01 \) and \( te'_3 = 2020-11-01 \) for all \( r \in R'_3 \); and \( R'_4 \) differs from \( R'_3 \) by assuming that \( inc'_4 = 55\% \) for all \( r \in R'_4 \).

We have that \(|P_1| = |P_2| = 59, |P_3| = |P_4| = 176, |R_1| = 42, \) and \(|R_2| = 24\) for \( 1 \leq i \leq 4 \). When \( P = P_1 \) (respectively, \( P = P_3 \)) and \( R = R_1 \), the actual number of relocated procedures with codes in \( P \) is at least 621 (respectively, 6506).

Representative results of our performance tests are provided below, where we report the actual numbers of variables and constraints passed to Cplex, neither counting variables whose values are known to be 0, nor constraints that express the lower bound of a variable.

Table 1 shows the number of variables and constraints together with execution times for Problems 3.1(a), 3.1(b), 3.2, 3.3(b) and 3.3(c), where parameters \( P, R, R' \) are set as shown in the second column.

| Problem | \( P, R, R' \) | #Variables | #Cons. | Time |
|---------|----------------|------------|-------|------|
| 3.1(a)  | \( P_1, R_1, R'_1 \) | 59473      | 67    | 3    |
| 3.1(b)  | \( P_1, R_1, R'_2 \) | 646759     | 239   | 38   |
| 3.2     | \( P_1, R_1, R'_3 \) | 1367857    | 312   | 151  |
| 3.2     | \( P_2, R_1, R'_3 \) | 1002961    | 312   | 97   |
| 3.3(b)  | \( P_1, R_1, R'_4 \) | 1367857    | 312   | 107  |
| 3.3(c)  | \( P_2, R_1, R'_4 \) | 1002961    | 312   | 109  |

Our method solving Problem 3.1(c) for parameters \( P, R, R' \) set to \( P_1, R_1 \) and \( R'_1 \) applies iterative deepening search, where:

• for computing the date \( te'_0 \) specified by (1), the corresponding Mip formulation uses 654 193 variables and 240 constraints;
• for checking the feasibility of the considered problem under the restriction that requires \( te' \) to be not later than the latest \( te \) (for \( r \in R_1 \)) more than 12 months, the corresponding Mip formulation uses 3 270 960 variables and 504 constraints;
• for subsequent iterations of the search, the corresponding Mip formulations use: 1 129 968 variables and 288 constraints; 1 367 856 variables and 312 constraints; 1 605 744 variables and 336 constraints.

In total, its execution took about 479 seconds, where most of the time was spent for the feasibility check.

The execution of our method for solving Problem 3.1(c) for parameters \( P, R, R' \) set to \( P_2, R_1 \) and \( R'_1 \) is analogous to the one for the case of \( P_1, R_1 \) and \( R'_1 \). It calls the Cplex solver 5 times (once for computing \( te'_0 \), once for checking the feasibility, and 3 times for iterative deepening steps), but each time with fewer variables (e.g., with only 1 369 872 variables for checking the feasibility). Totally, it took about 287 seconds.

When testing the scalability by changing \( P \) from \( P_1 \) to \( P_3 \), whose size is nearly 3 times bigger than the size of \( P_1 \), the execution time changes as shown in Table 2.

| Problem | Parameters | Time factor |
|---------|------------|-------------|
| 3.1(c)  | \( R = R_1, R' = R'_1 \) | 7.01         |
| 3.1(c)  | \( R = R_1, R' = R'_2 \) | 12.02        |
| 3.2     | \( R = R_1, R' = R'_3 \) | 2.97         |
| 3.3(b)  | \( R = R_1, R' = R'_4 \) | 104.34       |
| 3.3(c)  | \( R = R_1, R' = R'_4 \) | 5.45         |

Table 2. Results concerning scalability when changing \( P \) from \( P_1 \) to \( P_3 \). The third column shows the execution time multiplication factor.

| Problem | Parameters | Factor |
|---------|------------|--------|
| 3.1(c)  | \( R = R_1, R' = R'_1 \) | 2.6    |
| 3.2     | \( R = R_1, R' = R'_2 \) | 13.5   |
| 3.3(b)  | \( R = R_1, R' = R'_4 \) | 3.2    |
| 3.3(c)  | \( R = R_1, R' = R'_4 \) | 3.3    |

Table 3. Results concerning scalability when changing \( P \) from \( P_3 \) to \( P_4 \).

The third column shows the execution time multiplication factor.

Table 4. Results concerning time spent for constructing a model (\( t_1 \)) and finding a solution (\( t_2 \)).

| Problem | Parameters | \( t_1 \) (s) | \( t_2 \) (s) |
|---------|------------|--------------|--------------|
| 3.2     | \( P_3, R_1, R'_3 \) | 54           | 97           |
| 3.3(b)  | \( P_3, R_1, R'_4 \) | 163          | 1 862        |
| 3.3(c)  | \( P_3, R_1, R'_4 \) | 72            | 35           |
| 3.3(b)  | \( P_3, R_1, R'_4 \) | 218          | 118          |
When testing the scalability by changing $P$ from $P_2$ to $P_4$, whose size is nearly 3 times bigger than the size of $P_2$, the execution time changes as shown in Table 3. The results concerning time spent for constructing the model, denoted by $t_1$, together with time took by the CPLEX engine for finding a solution for the model, denoted by $t_2$, are shown in Table 4. In general, time spent for constructing the model is linear in the size of each of $P$, $R$ and $R'$.

One of the options of our implementation is to treat the considered MIP problems as linear programming problems, by allowing the variables $move_{p,t,i,r,t'}$ to take non-integer values and rounding their returned values. Using this option, the execution time increases only from 2.6 to 3.5 times for all of Problems 3.1, 3.2, 3.2(b) and 3.2(c) when changing $P$ from $P_1$ (resp. $P_2$) to about 3 times bigger $P_3$ (resp. $P_4$), and using $R = R_1$ and $R' ∈ \{R'_1, R'_2, R'_3, R'_4\}$ appropriately.

6. Conclusions

As a side-effect, Covid-19 pandemics resulted in a substantial decline of medical procedures performed in hospitals. In response to that phenomenon, a decision support is needed for planning the procedures’ relocation. In the paper we have formulated the corresponding problems, developed MIP models and verified their performance on real-world data using the CPLEX solver. Though MIP is generally NP-complete, experiments show that the developed models are computationally feasible. This makes them applicable in decision support systems.

In order to verify the feasibility of our solutions we have used data from Polish hospitals. However, the models are largely independent of the Polish healthcare system and can be applied in other circumstances as well. To our best knowledge, the obtained models and results have not been reported in the literature so far.

Although the models we developed have been primarily motivated by the Covid-19 pandemics, they may have applications in crisis management whenever healthcare industry is affected by epidemics, natural disasters or other circumstances resulting in substantial declines for medical services.

Last, but not least, as an illustrative range of medical procedures, the case of urology has been chosen and we have verified our models on urological data from Polish hospitals.

The models can be scaled in two main directions. First, they can cover larger territories employing parallel architectures. Second, one can adapt the models to other NCo specialties, including cardiology, neurology, etc. Here one could also relatively easily parallelize computations. However, modeling interactions among specialties would require more involved tools. Good candidates are Constraint Logic Programming interpreters [1, 4], possibly equipped with Machine Learning techniques for mining interactions from data and forecasting the demands for medical procedures in the considered future time intervals.

Acknowledgment: The research has been supported by the Polish Ministry of Science and Higher Education under project ProME “Predictive Modeling of the Covid-19 Epidemics”. We would also like to thank M. Niegódko, A. Powała, P. Regulski, A. Woźnica, P. Dunin-Kęplicz, M. Iwański and P. Wiśniewski for helpful comments and discussions.

References

[1] K. Apt and M. Wallace. Constraint Logic Programming using ECLIPS*. Cambridge University Press, 2006.
[2] A. Buecker, Y. Ageeva, V. Blanchard, J. Bloom, M.F. Candas, J. Chaves, G. Feng, A. Raman, and H. Schlenker. Optimization and Decision Support Design Guide Using IBM ILOG Optimization Decision Manager. IBM, 2012.
[3] A.Y. Chang, M.R. Cullen, and M. Harrington, R.A. Barry. The impact of novel coronavirus COVID-19 on noncommunicable disease patients and health systems: a review. J. Internal Medicine, 2020. doi: 10.1111/joim.13184.
[4] T. Frühwirth. Constraint handling rules. Cambridge Univ. Press, 2009.
[5] R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller, J.W. Thatcher, and J.D. Bohlinger, editors, Complexity of Computer Computations, pages 85–103. New York: Plenum, 1972.
[6] A. Miyawaki, J. Tomio, M. Nakamura, H. Ninomiya, and Y. Kobayashi. Changes in surgeries and therapeutic procedures during the COVID-19 outbreak: A longitudinal study of acute care hospitals in Japan. Annals of Surgery, 2020. doi: 10.1097/SLA.0000000000004528.
[7] Ch.H. Papadimitriou. On the complexity of integer programming. J. ACM, 28(4):765–768, 1981.
[8] M.L. Pinedo. Scheduling: Theory, Algorithms, and Systems. Springer, 5th edition, 2016.
[9] S.V. Sevastyanov, B.M.T. Lin, and H.-L. Huang. Tight complexity analysis of the relocation problem with arbitrary release dates. Theoretical Computer Science, 412(35):4536–4544, 2011.
[10] WHO. The Impact of the COVID-19 Pandemic on Noncommunicable Disease Resources and Services: Results of a Rapid Assessment. Geneva: World Health Organization, 2020.