A data-driven digital application to **support the capacity planning** of the COVID-19 vaccination process

Berend Markhorst, Renze Dijkstra, Daan Otto, Nina Malbasic, Tara Zver
Vrije Universiteit

Rob van der Mei
Department of Mathematics, Vrije Universiteit / Center for Mathematics and Computer Science

Dennis Moeke
Research Groep Logistics and Alliances, HAN University of Applied Sciences

In this paper an advanced decision support system (DSS) is presented which supports decision making regarding the capacity planning of the COVID-19 vaccination process. With the national 'vaccination priority list' as starting point, the DSS aims to minimize the per-class waiting-time with respect to the locations of the medical hubs (i.e., the vaccination locations) and the distribution of the available vaccines and healthcare professionals over the medical hubs over time. As the user is given the freedom to experiment with different starting positions and strategies, the DSS is ideally suited for providing support in the dynamic environment of the COVID19 vaccination process.

Introduction

The coronavirus pandemic has been holding the Netherlands and the world in its grip for almost a year now. Since the start of the outbreak, the virus has not only posed immense challenges to the Dutch healthcare system but also to its society. The economic, social, political and cultural effects have been (and still are) central topics of debate. And although some serious measures have been taken to minimize exposure and transmission of the coronavirus, widespread vaccination is believed to be a safe and critical step in ending the pandemic. Parties are working hard to create vaccines, the first of which are already available. But how do we get the available vaccines fast, carefully and responsibly from 'lab to arm'?

The production and delivery of COVID-vaccines is a complex logistic challenge. According to KPMG (2020), the last-mile delivery is one of the most critical parts of the COVID-19 vaccine supply chain. It is this last-mile challenge which is addressed in this paper. More specifically, a data-driven digital application to support the capacity planning of the vaccination process is presented and discussed. It should be noted that this decision support system (DSS) has been developed as part of a four-week educational project of the Business Analytics program of the Vrije Universiteit Amsterdam. For the end result, together with an instruction video, see https://lab-to-arm.com.

The remainder of this paper is structured as follows. In the next section, the problem is further explained and defined more formally. In section 3, the modelling approach behind the DSS is discussed. Section 4 presents the user interface of the DSS and describes the various features that are under control of the user. In the concluding section it is discussed how the DSS presented in this paper could aid decision-makers in addressing complex issues of priority setting and resource allocation regarding the vaccination process. Furthermore, opportunities for the further development of the DSS are outlined.

Problem description

When it comes to the streamlining of the vaccination, it is required to make a distinction between 'vulnerable persons' and 'less vulnerable persons'. This is because, from a logistics perspective, the needs and requirements differ vastly. Vulnerable persons, which have higher priority, will be vaccinated in their own homes, in hospitals or the nursing homes where they stay. Less vulnerable persons will be administered via large-scale vaccination centers, which could be set up in places like sport centers or stadiums. The focus of the contributions presented in this paper lies on the vaccination of less vulnerable persons via large-scale vaccination center, which from now on will be referred to as (medical) hubs.

The National Institute for Public Health and the Environment (RIVM) has published a priority list which assigns different groups of the Dutch society to different vaccines and vaccination moments. For example, persons between 18 and 60 years old without a medical condition have the lowest priority. Hence, different priority classes can be defined for the vaccinations at the medical hubs.

An important performance measure in this context is the per-class waiting time distribution. In other words: x procent of persons in priority class A have to wait y time units to be vaccinated. Note that these distributions may also be location-dependent. For example, the waiting times in the urban areas may be different than in rural areas. Obviously, the performance of the last mile of the vaccination program does not only depend on those priorities, but on other parameters as well, such as:

- The number and locations of the medical hubs.
- The available vaccination capacity of a medical hub.
- The number of healthcare professionals (authorized to administer the vaccine).
- The number of available vaccines per medical hub.

In this paper, we present an optimization model that optimizes the aforementioned performance measure of per-class waiting time. This is done by determining the optimal placement of medical hubs across the Netherlands, as well as the division of vaccines and healthcare personnel between those hubs. Secondly, with the optimization model as core input, a user-friendly DSS is presented which provides the user with the possibility of conducting what-if analysis. As part of this what-if functionality, the DSS requires the user to define several input parameters of which an overview can be found in Section 4.

Modelling approach

The problem can roughly be divided into the following processes:

1. Finding the 'optimal' locations of the medical hubs.
2. Finding the 'optimal' distribution of the medical capacity (i.e., the available vaccines and nurses) over the medical hubs.
3. Simulation of the vaccination process.

The second and the third can be combined to find the optimal distribution of medical capacity over the hubs.

It should be stressed that the model (underlying the DSS) is not 'optimizing' the vaccination process, as it involves a lot of subjectivity. For example, different persons might answer differently to questions such as: To what extent should healthcare workers get more prioritization?. It is difficult to answer such questions strictly mathematically. Therefore, the user is given the freedom to experiment with different starting positions and strategies (i.e., conduct 'what-if' analysis). In addition, with respect to the development of the algorithms trade-offs had to be balanced between computation time and accuracy. Hence, the DSS should be sufficiently accurate and at the same time provide the user with a smooth interactive experience.

Hub placement

When it comes to the placement of the medical hubs, the facility location problem, (e.g., Adeleke & Olukanni, 2020), set covering problem (e.g., Daskin, 2011), and COVER-CAP (Lee et al., 2009) are well known formulations (and extensions) of the problem at hand. As these approaches are all based on integer linear programming (ILP) the state space suffers from the curse of dimensionality. Consequently, the computation time of these approaches are prohibitively long given the large number of constraints and decision variables as they occur in practice.

As the problem at hand could not be solved to optimality within an acceptable time frame, a fast yet accurate heuristic was required. The developed heuristic resembles the divide-and conquer structure discussed by Cormen et al. (2009). Based on this structure, we partitioned the problem-solving process into two steps. In the first step, we assume that each municipality has its own hub. This step is referred to by Cormen et al. (2009) as the dividing step. In the second step, we try to merge hub locations. More specifically, hub locations are merged if the distance constraints can be met; otherwise the municipality will get its own hub. The merging of hub locations corresponds with the so-called conquer-step of Cormen et al. (2009). When executing this algorithm, it is ensured that each municipality is covered. See Figure 1 Below for the pseudo-code of the hub-placement algorithm.

```
Algorithm 1 Compute locations of the hubs
every municipality has its own hub.
while not every hub is done do
  sort the hubs in increasing order on number of inhabitants
  select the smallest hub that is not done yet (e.g. hub A)
  find the nearest hub to the smallest hub (e.g. hub B)
  if the maximal distance constraint is violated then
    hub A is done
  else
    if distance constraint between centre hub A and all municipalities in hub B are not violated then
      merge municipalities hub B in hub A
      remove hub B from hub list
    else if distance constraint between centre hub B and all municipalities in hub A are not violated then
      merge municipalities hub A in hub B
      remove hub A from hub list
    else
      hub A is done
    end if
  end if
end while
```

Figure 1 Algorithm 1: Pseudo-code for the placement of hubs

Two limitations should be mentioned with respect to the hub-placement algorithm:

1. As we use heuristic Euclidean distances between the centroids of municipalities, the 'true' travel distance, which can be calculated by using route planners (depending on the travel modality), may exceed the maximum travel distance.

2. The heuristic does not take the costs concerned with opening a hub into account. This was a design decision, as these costs were not known beforehand. However, having the algorithm to minimize the number of hubs does work in favor of lowering the hub opening costs.

Distribution of medical capacity

Regarding the distribution of medical capacity, it is assumed that the order in which priority groups are vaccinated is given and can also be altered by the end-user. Regarding the allocation of medical capacity, two options were implemented:

Equal distribution of vaccines across the hubs

The available number of vaccines per day is equally distributed across the hubs that are not finished with vaccinating yet. Thus, if on a given day, H hubs are not finished with vaccinating, and there are V vaccines available for that day, then each of those H hubs will get the same number of vaccines, namely:

$$\lfloor \frac{\boldsymbol{V}}{\boldsymbol{H}} \rfloor$$

The advantage of this method is that each hub will receive the same number of vaccines, independently of their size. As a result, smaller hubs will be finished with vaccinating earlier than larger hubs. Consequently, larger hubs will eventually get more vaccines per day once the number of finished hubs increases and are not considered in the division of vaccines anymore. A major disadvantage of this method is that, during the early stage, densely populated areas will not get more vaccines, which increases the risk of a local COVID-outbreak.

Proportional allocation of vaccines over hubs

This strategy is based on the number of susceptible persons. For this strategy, the number of non-vaccinated persons (i.e., susceptible persons) in the whole Netherlands, S_NL, and the number of non-vaccinated persons for each hub, S_h, are determined. The number of available vaccines per day is denoted as V. To calculate the number of vaccines that each hub gets per day, the following formula is used:

$$\lfloor \frac{\boldsymbol{S_h}}{\boldsymbol{S_{NL}}} * \boldsymbol{V} \rfloor$$

Over time, every hub will get the same number of vaccines, as the proportional distribution has not been altered. As a result, every hub will be finished vaccinating almost simultaneously. An advantage of using this proportional allocation is that the areas that are more likely to endure local outbreaks (due to a high number of non-vaccinated persons) will receive more vaccines. This would result in the risk of the local outbreaks being lower for these areas.

The two aforementioned allocation strategies are implemented in the algorithm for the allocation of vaccines and nurses across the hubs.

Vaccine allocation

The pseudo-code of the vaccine allocation algorithm is shown in Figure 2. The implementation consists of two parts: the allocation and the number of shots required. Being able to deal with both one and two-shot vaccines requires more advanced algorithms. For example, once a person is vaccinated for the first time, another shot from today's supply should be reserved for the person to be given later. To support this additional feature, the time period between the first and second shot has been added as an additional input parameter. In this way, it is ensured that there enough vaccines available to provide persons with their second shot. The reason for choosing this approach is to make the model more realistic as it resembles the strategy of the Dutch government.

Algorithm 2 Division of vaccines

With some divisions, it is necessary to floor (**) the resulting numbers.

```
for every hub do
  if allocation is proportional then
    Total left over: compute the national number of people that are left over from all priority classes
    Hub left over: compute the number of people that are left over from all priority classes on a hub-level
    if Hub left over is 0 then
      Hub is done vaccinating new people.
    else
      (**) Amount of vaccines: hub left over / total left over * number of vaccines available
    end if
  else if allocation is equal then
    Left over hubs: compute the number of hubs that are still busy vaccinating new people
    if hub is still busy vaccinating new people then
      (**) Amount of vaccines: number of vaccines available / left over hubs
    else
      Hub is done vaccinating new people.
    end if
  end if
end for
```

Figure 2 Algorithm 2: pseudo-code for the allocation of vaccines

Nurse allocation

Figure 3 shows the pseudo-code of the nurse allocation algorithm. Before applying the equal and proportional approach, which is similar to the vaccine allocation, a distinction should be made between a single- or double-shot vaccine. In case of a double shot vaccine, the algorithm reserves sufficient capacity for the second shot. Once the required 'second shot capacity' has been determined, the remaining number of nurses gets divided over the hubs either proportionally or equally.

```
Algorithm 3 Division of nurses
With some divisions, it is necessary to floor (**) the resulting numbers and with others to ceil (*).
if two shots are necessary then
  for every hub do
    Second shot: check how many people come for a second shot.
    (*) Minimal amount of nurses: second shot / (8 hours * vaccination speed per hour per nurse)
  end for
end if
Compute the remaining number of available nurses.
for every hub do
  if allocation is proportional then
    Total left over: compute the national number of people that are left over from all priority classes
    Hub left over: compute the number of people that are left over from all priority classes on a hub-level
    (**) Amount of nurses: hub left over / total left over * number of nurses available
  else if allocation is equal then
    Left over hubs: compute the number of hubs that are still busy vaccinating new people
    if hub is still busy vaccinating new people then
      (**) Amount of nurses: number of nurses available / left over hubs
    else
      Amount of nurses: 0
    end if
  end if
end for
```

Figure 3 Algorithm 3: pseudo-code for the allocation of nurses

The vaccination process

In the previous sections, the approach for determining the hub locations and allocation of nurses and vaccines across these hubs has been described. In this section, it is explained how a 'vaccination-epoch' is simulated, until all willing members of all priority classes have been vaccinated. Note that the persons not willing to get vaccinated will not be considered. To compute the number of persons that can get a vaccine, the total vaccination time is computed as follows:

$$min\ (\#nurses * \#vaccines\ per\ nurse\ per\ hour * 8\ hours, \#available\ vaccines)$$

The reason for this approach is that if there are less vaccines available than the nurses can administer, the vaccines are the bottleneck. If not, then 'nurse capacity' is the bottleneck. Note that if a person requires two doses of the vaccine, the number of available vaccines is divided by two, in order to ensure availability for the second shot. This process is repeated until the first priority group is fully vaccinated. After that, the vaccination process of the second priority group starts. In the meantime, the model keeps track of the number of persons getting their first or second shot, when they received the shot and how many vaccines and nurses were needed per hub. This process is repeated until all applicable persons, willing to get the vaccine, have been vaccinated.

The verification of the correctness of the hub placement is performed via the placement of circles, with a radius of the maximum travel distance, on the chosen hubs. If the areas of all circles cover the centroids of all the municipalities, we can conclude the algorithm has found a feasible solution. An example is provided in Figure 4. The distribution of vaccines and nurses and its effect on the vaccination timeline has been verified numerically (comparison between numerical computations and the algorithm's outcomes).

Figure 4 Verification of the placement of hubs by checking the maximum travel distance constraint (in this example 60 km).

Decision Support System

In this section we present a DSS which can be used to evaluate alternative strategies by allowing the user (in a user-friendly way) to change its input parameters. User-friendliness was a key element during the development of the dashboard of the DSS. Hence, in the development of the dashboard particular attention has been paid to the clarity, intuitiveness and response time. As mentioned in Section 2, the key in this context is the per-class waiting time distribution, defined as *'x procent of persons in priority class A have to wait y time units to be vaccinated'.*

Input parameters

To allow the user to interactively explore a wide variety of what-if scenarios, several input parameters must be tuned to match the required what-if situation. The DSS requires the following input parameters to be specified by the user:

- **The number of provided vaccines per period:** To include uncertainty regarding the number of available vaccines, one can dynamically fill in the number of available vaccines per day per period. Thus, different amounts can be used for various dates.

- **The number of vaccines per hour per nurse:** The (average) number of vaccines that can be administered by one nurse per hour, given that a working day is 8 hours long.
- The number of nurses per day: The total number of available nurses per day. These nurses will be distributed across the hubs according to the selected allocation strategy.
- **The allocation of nurses and vaccines:** Two different allocation strategies can be chosen from: 'equal' and 'proportional.' By choosing the option 'equal' the available number of nurses and vaccines will be equally distributed across all hubs. On the other hand, by choosing 'proportional' the available number of nurses and vaccines will be divided between the hubs according to the proportion of inhabitants that still need to be vaccinated in that hub.
- **The vaccination strategy prioritization:** The Dutch government has defined priority classes. For this model specifically, only three have been considered: healthcare workers, non-vulnerable elderly (65+), and non-vulnerable adults (18-65). However, to account for the continuously changing order of vaccination prioritization, this input parameter has been implemented. It enables the user to switch the order in which these three priority classes should be vaccinated.
- **The number of vaccine shots:** The user can select whether one or two shots (doses) are required. As the majority of available vaccines in the Netherlands require two shots, the latter option is set as default value.
- **The number of days between the shots:** This is an essential input parameter, as the number of days between two shots can vary between vaccines. In addition, it should be mentioned that once a person receives the first shot (of a two-shot vaccine), the second shot is instantly saved and is not given to anyone else. This ensures that each person who is vaccinated once gets the other shot, regardless of the storage of available vaccines.
- **The maximum acceptable travel distance**: The maximum distance (in kilometers) that persons are willing to travel for getting their vaccine.
- **The coverage ratio:** The proportion of the Dutch population that should be vaccinated.
- **The willingness to get vaccinated:** The proportion of persons that is willing to get vaccinated (per priority class).

It should be noted that each input parameter has a predefined default value.

Output

The dashboard of the DSS provides the user with the following information:

- The total number and location of the (medical) hubs
- The overall waiting time
- The waiting times per priority class (healthcare workers, elderly, and adults)

Moreover, the user can gather insights from different graphs shown by using the tabs 'Overview' and 'More detailed information'.

Overview tab

The first three visuals discussed in this tab cover the guidelines for policy makers to decide where and how many medical hubs should be placed. These visuals are closely related to the maximum travel distance parameter, which the user can adjust in the input section.

- **Location of the hubs:** This graph shows the placement of the medical hubs in the Netherlands, where each point denotes a single hub. The user can hover over the hubs, which will provide each medical hub's location.
- **Area of the hubs:** The user is provided with the option to see which municipality belongs to which medical hub, as multiple municipalities can belong to a single hub (see Figure 5.). In addition, by hovering over the map the user can see more detailed information about each municipality.

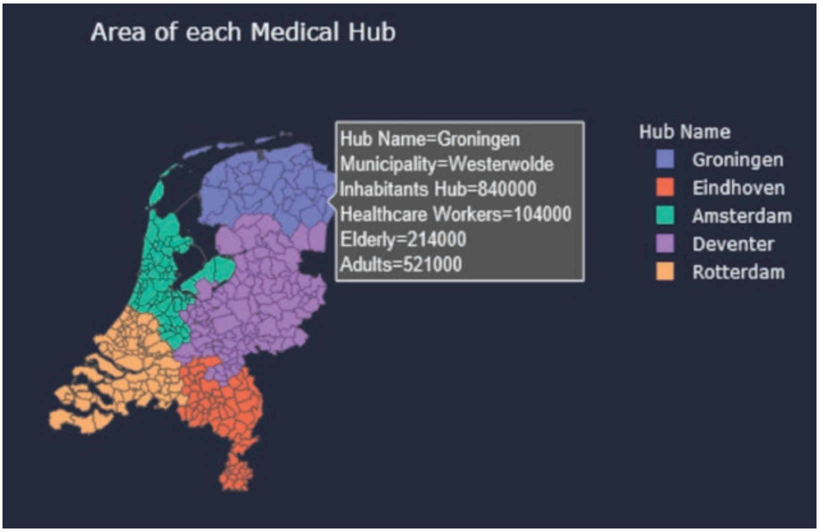

Figure 5 Overview of the location and number of medical hubs.

- **Impact travel distance:** The user can see the effect that the maximum travel distance has on the number and location of the medical hubs.

Furthermore, the bottom two visuals in the 'Overview'-tab provide more insight into the actual vaccination process.

- **Vaccination process over time (graph):** The graph shown in Figure 6 provides detailed information on the fraction of vaccinated persons over time for each priority class. The exact information on when a particular priority class is finished with vaccinating is also given by the KPI's at the top of the page.
- **Vaccination process over time (animation):** Closely related to the graph mentioned above, an animation of the vaccination process over time can be found.

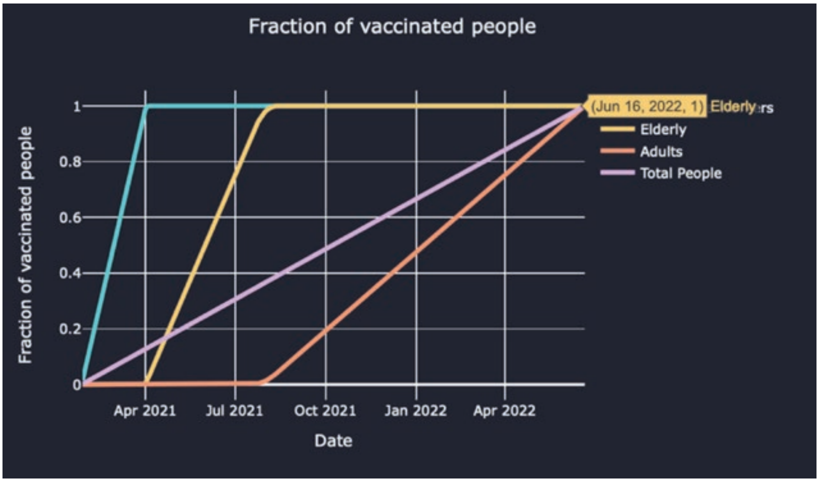

Figure 6 Fraction of vaccinated persons over time for each priority class.

More detailed information tab

The dashboard's final tab provides the user with insight on how the chosen starting position and vaccination strategy effects the allocation of vaccines and nurses across the hubs (over time).

- **Division of nurses/vaccines:** By clicking the 'Select division' button, the user can choose between nurses and vaccines. The visual (see Figure 7) shows how the vaccines are allocated across the medical hubs over time.

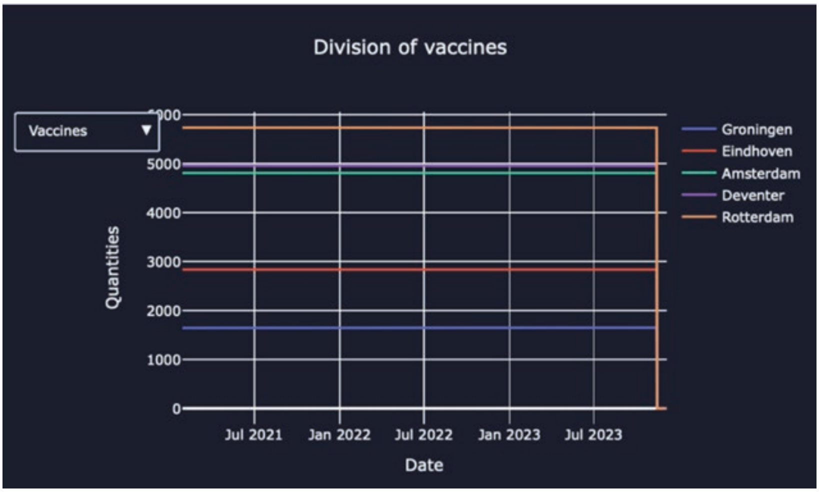

Figure 7 Allocation of vaccines across the medical hubs over time.

- **The fraction of vaccinated persons:** Figure 7 shows the division of vaccines across hub over time on an aggregate (i.e., national) level. However, the dashboard also provides the user with in-depth information on hub level.
- **The fraction of persons with the first shot (active):** This graph, shown in Figure 8, will only appear when the option 'two shots' is selected. It shows, for each priority class, the fraction of persons who have received their first vaccine shot over time on a national level.

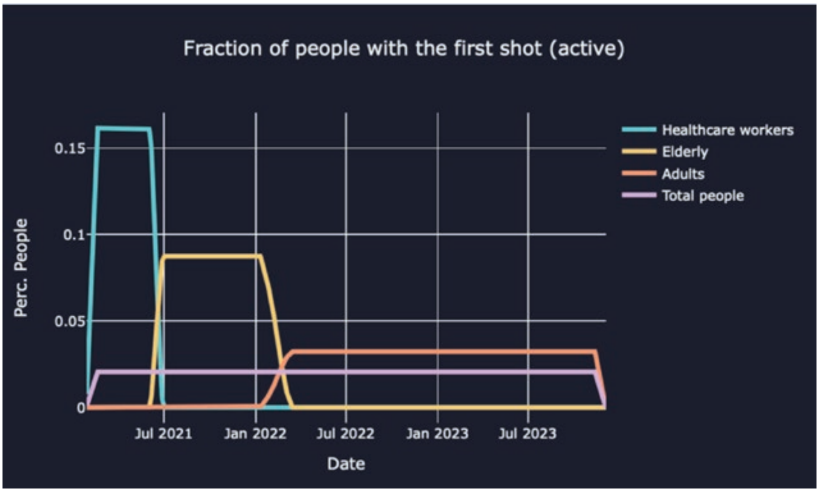

Figure 8 The fraction of persons who have received their first vaccine shot over time on a national level (for each priority class).

Conclusion and discussion

In this paper an advanced decision support system (DSS) is presented which supports decision making regarding the capacity planning of the COVID-19 vaccination process. With the national 'vaccination priority list' as the starting point, the DSS aims to minimize the per-class waiting time with respect to the locations of the medical hubs (i.e., the vaccination locations) and the distribution of the available vaccines and healthcare professionals between the medical hubs (over time).

As the user is given the freedom to experiment with different starting positions and strategies, the DSS is ideally suited for providing support in a continuously changing environment.

Besides providing support for the short-term (dynamic) planning, the DSS can assist policy makers in getting a better understanding the possible consequences of (restricting) the availability of key-resources (e.g., vaccines, vaccination capacity and number of medical hubs).

Despite being a powerful, *some important limitations should be* acknowledged. Firstly, transportation and storage constraints are not considered. Secondly, the applied approach assumes a smooth vaccination process.

Literature

Adeleke, O. J., & Olukanni, D. O. (2020). Facility location problems: models, techniques, and applications in waste management. Recycling, 5(2), 10.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms. MIT press.

Daskin, M. S. (2011). Network and discrete location: models, algorithms, and applications. John Wiley & Sons.

KPMG (2020), Five ways to optimize the COVID-19 vaccine supply chain, Retrieved from:https://assets.kpmg/content/dam/kpmg/xx/pdf/2020/12/five-ways-to-optimize-the-covid-19-vaccine-supply-chain.pdf.

Lee, E. K., Chen, C. H., Pietz, F., & Benecke, B. (2009). Modeling and optimizing the public-health infrastructure for emergency response. Interfaces, 39(5), 476-490.