

# Development of a Web-based Application for Optimal Allocation of COVID-19 Vaccine

John Frederick D. Tapia<sup>a,\*</sup>, Joshua Manzano<sup>b,c</sup>, Ian Benedict Ona<sup>b,c</sup>, Unisse Chua<sup>b,c</sup>

<sup>a</sup> Chemical Engineering Department, De La Salle University – Manila, 2401 Taft Avenue, Malate, Manila, Philippines, 0922

<sup>b</sup> Data Science Institute, De La Salle University – Manila, 2401 Taft Avenue, Malate, Manila, Philippines, 0922

<sup>c</sup> Center for Complexity and Emerging Technologies, De La Salle University – Manila, 2401 Taft Avenue, Malate, Manila, Philippines, 0922

[john.frederick.tapia@dlsu.edu.ph](mailto:john.frederick.tapia@dlsu.edu.ph)

The COVID-19 pandemic posed a huge impact to the global economy and on the environment. Recent developments in battling the pandemic include the availability of vaccines for public use. The distribution of vaccines requires systematic planning as supplies to a given region can be limited. Process integration techniques such as mathematical programming can be applied to develop solutions for systematic planning of COVID-19 vaccine distribution. To enable the use of mathematical programs for policymakers, a graphical user interface is needed. In this study, a web-based application (app) for allocating vaccines is presented, utilizing non-linear programs to optimally distribute vaccines to different population groups. Two models are incorporated in the app: (1) a model that minimizes the number of vaccines required to slow down the spread of the disease, i.e., reduce the reproductive number to less than one and (2) a model that determines the optimal allocation at the minimum reproductive number. The optimization models for the app were written in Python using the Pyomo package while the graphical user interface is written in JavaScript with the use of the React library. The optimization models are accessed through an API served using a Python web server utilizing the web framework Flask. A case study is used to illustrate how the models work in the web application.

## 1. Introduction

The COVID-19 pandemic has been an unanticipated global issue that affects not only the health sector but also the economy and education. The pandemic has caused closure of schools, establishments, and businesses around the world due to imposition of lockdowns to prevent the spread of the disease. The lockdowns are complemented with the development of drugs and vaccines. With its recent development, vaccines are being rolled-out to different countries. In the United States, it effectively decreases case counts and relaxed certain restrictions (Mendez and Rattner, 2021). In United Kingdom, no new deaths have been recorded since the vaccine roll-out (Marsh and Gayle, 2021). These nations have been able to effectively roll out vaccine supplies since these are produced locally: BNT162b2 mRNA vaccine (Pfizer/BioNTech) in the United States and ChAdOx1 nCov-19 vaccine in the United Kingdom. In developing countries where vaccine supplies are needed to be procured internationally, limited quantities can be acquired from time to time. The large difference in vaccination rates has been reported by Matthieu et al. (2021) where most developing countries are inoculating just 5-10% of its population. Thus, systematic allocation of vaccines is needed to be done to minimize the spread of the virus and the fatalities caused by the disease. This approach can minimize the environmental impacts of vaccine production and distribution through efficient allocation to increase human lives saved (Jiang et al., 2021). Mathematical tools are developed for vaccine allocation, however, the use of it for a non-expert perspective will be challenging. In this study, an optimization model for allocation for COVID-19 vaccines is implemented as a web application or web app.

Process integration techniques has been applied for allocation of resources in health care sector. An optimization model has been developed for the allocation of hospital resources for COVID-19 (Sy et al., 2020).

This study mainly includes the allocation of anti-viral drugs to patients with varying degree of severity. Fair use allocation based on needs per region (Munguia-Lopez and Ponce-Ortega, 2021) and dynamic approach to vaccine allocation for single and double doses (Matrajt et al., 2021) were developed for COVID-19. A model developed by Enayati and Ozaltin (2020) involves the vaccine allocation for influenza, considering of the dynamics of the disease. The model involves extracting information from the susceptible exposed infected recovery (SEIR) model to map the infection rate between population groups. These models are implemented in optimization software and are mainly used by an expert. Barriers from using these models includes the requirement of a lot of data, lack of scalability and lack of interdependence between cases in different regions. A user-friendly application is needed for health professionals, especially with little background on operations research, to use these models.

In this study, the implementation of the vaccine allocation is done through a publicly available web application. The model is written in Python using the Pyomo module (Hart et al., 2011). This module is used in different applications such as implementing decision tools for aircraft fleet maintenance (Pereira et al., 2021), heat exchange networks (Short and Isafiade, 2020) and tool for emulsion preparation (Torres et al., 2020). Implementing optimization tools in Pyomo allows the integration of other Python modules such as web development (e.g., Flask) and visualization tool (e.g., Matplotlib) to create a user-friendly application. Thus, user input and visualization of results are possible.

## 2. Mathematical formulation

Two non-linear models are implemented in this study: (1) minimizing the reproductive number and (2) minimizing the number of vaccines required to bring the reproductive number below acceptable level. These models are adapted from Enayati and Ozaltin (2020) and modified to generate two models that can be selected depending on the need of the user. The optimization problem involves  $m$  population groups and  $q$  types of available vaccines. Each group is characterized by the number of individuals while each vaccine is characterized by its efficacy. The approach considers the dynamics of the disease, which can be represented by the next-generation matrix. This matrix involves the number of infections a particular group can contribute to new cases. Then, the models can be solved one at a time, depending on the available information. Minimizing the reproductive number entails allocation of the available vaccines to minimize the spread. On the other hand, minimizing the required vaccines can be set to slow down the spread of the disease.

### 2.1 Minimizing Reproductive Number

The objective function is to minimize the spread of the virus, represented by its reproductive number,  $R$ :

$$\min R \quad (1)$$

subject to the following constraints:

$$\sum_i v_i = R \quad (2)$$

$$(1 - \sum_p H_p f_{ip})(\sum_j K_{ij} v_j) \leq R v_i \quad \forall i \quad (3)$$

$$\sum_i f_{ip} \leq 1 \quad \forall p \quad (4)$$

$$v_i \geq \min_j (K_{ij}) \quad \forall i \quad (5)$$

$$\sum_i \sum_p f_{ip} n_i \leq F_{avail} \quad (6)$$

The dynamics of the disease can be described by the next generation matrix,  $K_{ij}$  which indicates how many infections an individual from group  $i$  will be incurred in group  $j$ . The eigenvalue  $v_i$  is related to the reproduction of group  $i$  in the whole population. The reproductive number,  $R$ , is in turn the sum of these eigenvalues, given by Eq(2) while the dynamics of the disease is given by Eq(3). The first factor in Eq(3) considers the effectiveness of the vaccines to minimize the spread of the virus. The efficacy of vaccine  $p$ , is given by  $H_p$ . The decision variable is the fraction of population of group  $i$  to be administered with vaccine  $p$ ; it is represented by  $f_{ip}$ . The sum of this fractional allocation should be less than or equal to one for all population groups, as stated in Eq(4). The number of individuals vaccinated should not exceed the available number of vaccines. Non-trivial allocation can be generated using the constraint given by Eq(5), allowing for the minimum eigenvalue to exceed the smallest value of  $K_{ij}$  among population groups  $j$  in each population group  $i$ . This is added due to the nature of its global optimum that sets all eigenvalues to zero without this constraint. Eq(6) limits the allocation to the total available vaccine,  $F_{avail}$ .

## 2.2 Minimizing Vaccine Requirement

The objective function for this model is to minimize the vaccine requirement:

$$\min \sum_i \sum_p f_{ip} n_i \quad (7)$$

subject to the constraints given by the Eq(3) to Eq(5). Eq(2) is modified such that the reproductive number,  $R$ , is set to a value equal to 1.

$$\sum_i v_i = 1 \quad (8)$$

Instead of setting a fixed number of available number of vaccines, this is set as the objective function given by Eq(7). The implementation of both models will be given in the next section.

## 3. Software Implementation

For easy access without having to install the application, a web-based application was designed using React and Python. As a web application, policymakers and other potential users would be able to just access the website through any browser. The infrastructure of the application can be illustrated in Figure 1. The optimization model is written in Python with the aid of the Pyomo package and the interface uses the React JS library. The user inputs will depend on the model used, however, the next-generation matrix and the population distribution in each group is needed to be specified. It will be passed through the web framework implemented using Flask and to the variables used in the Python program. Finally, the solution will be displayed back to the web interface.

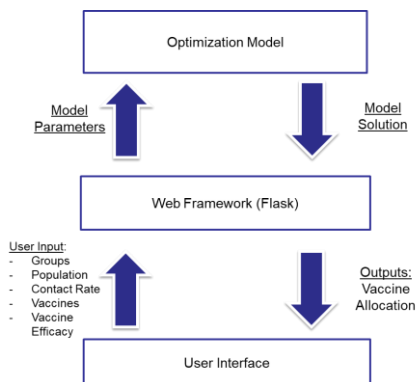


Figure 1: Overview of the framework for the COVID-19 vaccine allocation web app.

The web application was designed with a Model–view–controller pattern in mind, with the Model presented as an application programming interface (API). This allows the application to not be limited to a specific user interface and permits other external developers to conduct their own testing and create their own user interfaces querying the API.

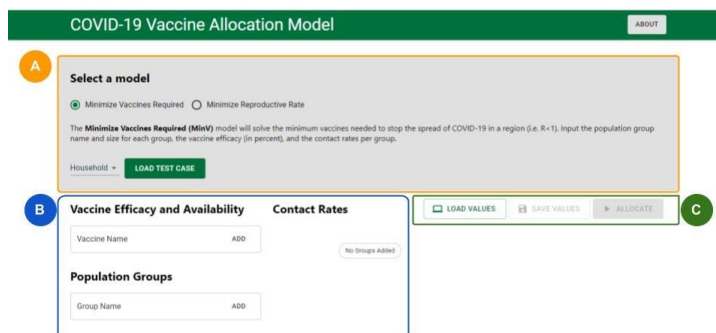


Figure 2: Starting screen of the COVID-19 Vaccine Allocation web application.

React was used for the development of the client-side web application. React is an open-source JavaScript library for building reactive user interfaces. The application was componentized into the main form, the footer, and the *About* section. The main form starts with a header box that allows the user to select between two of the

available models along with a brief description (See Figure 2A). It also allows the user to select from a pre-made set of test cases that pertain to common population group and contact rate examples: *Household*, *School*, *Workplace*, and *Public*. The next-generation matrices for these test cases were adapted from Liu et al. (2020). Alternatively, the user can also choose to manually input the data within the three input sections of the form which pertain to *Vaccine Efficacy*, *Population Groups*, and *Contact Rates* (See Figure 2B). The contact rates represent the values for the next-generation matrix. There are also buttons on the upper-right side of the form that allow the user to save the values into a comma-separated values (csv) file and load from a file of the same template (See Figure 2C). Beside these buttons is the *Allocate* button which will trigger an API call to the server and receive the calculated allocation values. Once these values are received, it will display them in a bar graph below the form. Currently, the application has undergone functional testing to ensure that the model and the error validations are fully working. However, usability testing with the target users have not yet been conducted. The program is available in this website: <https://dlsudatasci.github.io/covidapp/>.

#### 4. Illustration/Demonstration

To develop the backend of the application, Python was used as the main programming language due to the availability of the Pyomo package which is mainly used to develop optimization models. To connect the user interface with the functionalities of the optimization models, a Python web server utilizing the web framework Flask was used to serve the requests given from the client-side. The backend was modularized into three main components: one containing the model and solver logic, another containing the input validation logic, and another containing the web server logic as an API which receives the requests. The rationale behind the modularization is to allow for extensibility and flexibility. This makes adding multiple new models and changing of validation logic easy and opens the application for restructuring and addition of new components. Two single objective models are included in the web application, to allow flexibility in terms of the user's specific need.

The application is demonstrated using a case study. The typical vaccine efficacy based on the results of the Phase III clinical trials is at least 50% (Cheng et al., 2021) which is used in this case. Table 1 shows the details in the test case used, indicating the next generation matrix. The next generation matrix can be interpreted as the number of individuals that an individual in a particular group can infect in another. For instance, Table 1 shows that a value of 0.6 within Group A means that for every 100 cases in this group, 60 additional cases may be added in Group A in the next generation. The term "next-generation matrix" is changed to *Contact Rates* in the web application. Five groups are used for the demonstration segregated by age. For the model that minimizes the reproductive number, the available supply is good for 187 people. The population of each group are as follows. Group A has 77 individuals while B has 241 individuals. Groups C to E have 375, 204 and 103 individuals, respectively.

Table 1: Next generation data for the illustrative case study

	A	B	C	D	E
A	0.6	0.1	0.1	0.1	0.1
B	0.2	1.7	0.3	0.2	0.2
C	0.4	0.3	0.5	0.4	0.3
D	0.2	0.1	0.3	0.2	0.1
E	0.1	0.1	0.1	0.1	0.1

From the starting screen of the application, the user can first select the desired model in the header box. Afterwards, they can input the vaccine information under the *Vaccine Efficacy and Availability* section of the form (See Figure 3A). Each vaccine can be given a name, efficacy rate, and the number of available vaccines. The user can then input multiple entries under the *Population Group* section (See Figure 3B). Each group can be given a name and the population count. Every group added will also insert a corresponding row and column under the *Contact Rates* section (See Figure 3C) wherein users can input the values for the next generation matrix. Figure 3 shows a filled-out form using the test case. The values in the above table can be copied and pasted into this section.

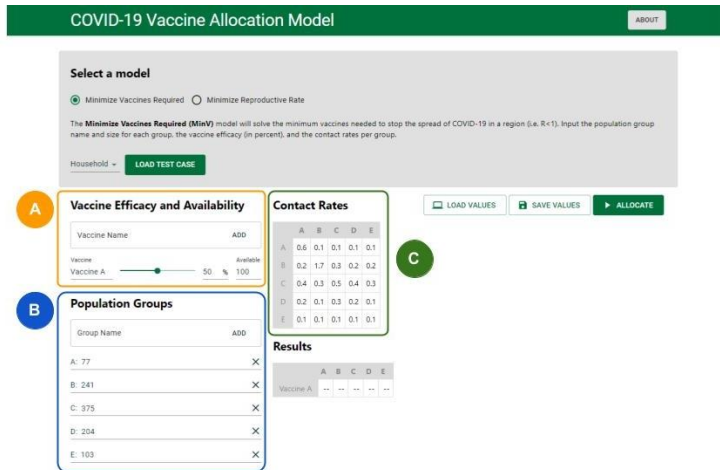


Figure 3: A sample scenario fully entered onto the web application.

After the form has been filled up, the user can click on the *Allocate* button on the upper-right hand corner of the form to run the selected model with the inputted data. Note that the application will notify the user if a required value is missing or if the model flags an error, as shown in Figure 4.

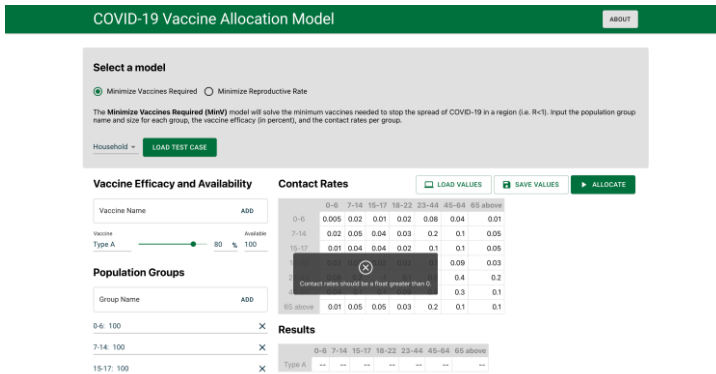


Figure 4: A sample error message when contact rates entered are not within the valid ranges.

If no errors are encountered, the results will then appear in a table on the lower-right side of the form and in a bar graph below the form as shown in Figure 5. The bar graph is generated using the results and will show a stacked bar chart when there are multiple types of vaccines present.

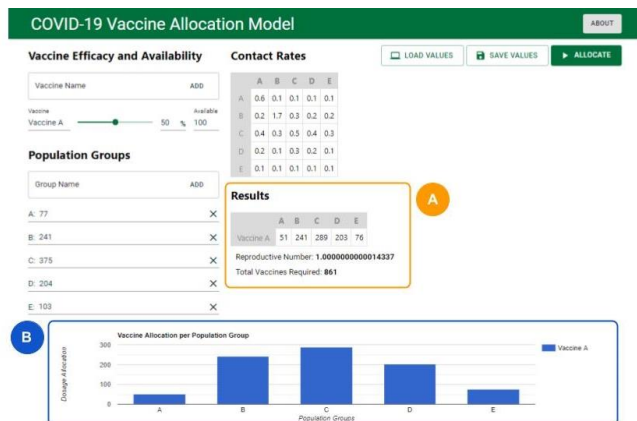


Figure 5. Model results are presented in a table below the Contact Rates with a simple bar graph to show the distribution of the vaccine allocation.

The solution for the test involves the need to allocate the vaccines such that all the available vaccines is given to Group B. This involves vaccinating 77.6% of the population of Group B to reduce the reproductive number to 1.3. However, if we want to minimize the spread of the virus, the required number of vaccines should be good for 861 people or more. The allocation involves allocating 33.6% of the available vaccine to Group C and 28% to Group B. These two groups are given with the highest allocation while Group A only receives 6% of the available vaccines: the lowest among the groups. The software implementation allows the user to provide information to the software in a user-friendly manner. The interface allows the user to understand which information is needed to gain insights on systematic allocation of COVID-19 vaccines.

## 5. Conclusions and Future Work

A web application tool has been developed to implement two optimization models for vaccine allocation considering the infection rate between population group. The web application is written in Python where the optimization model is written using the Pyomo package. The integration of the components is done through Flask while the interface is written using React JS library. The web application allows better access to systematic decision tool for vaccine allocation, especially if the vaccine supply is limited. A case study is illustrated, where the use of application is demonstrated. Future work includes documentation of the API for developers to work on the model.

### Acknowledgements

The authors wish to acknowledge DLSU Animo Labs Foundation Inc. for funding this research.

### References

- Cheng, H., Peng, Z., Luo, W., Si, S., Mo, M., Zhou, H, b, Xin, X., Liu, H., Yu, Y., 2021, Efficacy and safety of covid-19 vaccines in phase III trials: A meta-analysis, *Vaccines* 9(6), 582.
- Enayati, S., Ozaltin, O., 2020, Optimal influenza vaccine distribution with equity, *European Journal of Operational Research*, 283, 714–725.
- Hart, W.E., Watson, J.-P., Woodruff, D.L., 2011, Pyomo: Modeling and solving mathematical programs in Python, *Mathematical Programming Computation*, 3(3), 219–260.
- Jiang, P., Klemeš, J.J., Fan, Y.V., Fu, X., Tan, R.R., You, S., Foley, A.M., 2021, Energy, environmental, economic and social equity (4E) pressures of COVID-19 vaccination mismanagement: A global perspective, *Energy*, in press, 121315.
- Liu, Y., Gu, Z., Xia, S., Shi, B., Zhou, X.-N., Shi, Y., Liu, J., 2020, What are the underlying transmission patterns of COVID-19 outbreak? An age-specific social contact characterization, *EClinical Medicine*, 22, 100354.
- Marsh, S., Gayle, D., 2021, Zero daily COVID deaths announced in UK, *The Guardian*, <[www.theguardian.com/world/2021/jun/01/zero-daily-covid-deaths-announced-in-uk](http://www.theguardian.com/world/2021/jun/01/zero-daily-covid-deaths-announced-in-uk)> accessed 14.06.2021.
- Mathieu, E., Ritchie, H., Ortiz-Ospina, E., Roser, M., Hasell, J., Appel, C., Giattino, C., Rodes-Guirao, L., 2021, A global database of COVID-19 vaccinations, *Nature Human Behaviour*, DOI: 10.1038/s41562-021-01122-8.
- Matrajt, L., Eaton, J., Leung, T., Dimitrov, D., Schiffer, J.T., 2021, Optimizing vaccine allocation for COVID-19 vaccines shows the potential role of single-dose vaccination, *Nature Communications*, 12(1), 3449.
- Mendez, R., Rattner, N., 2021, More than 50% of Americans got at least one Covid vaccine shot as U.S. cases fall further, *CNBC*, <[www.cnn.com/2021/06/01/covid-19-cases-deaths-vaccinations-daily-update.html](http://www.cnn.com/2021/06/01/covid-19-cases-deaths-vaccinations-daily-update.html)> accessed 14.06.2021.
- Munguia-Lopez, A.C., Ponce-Ortega J.M., 2021, Fair allocation of potential COVID-19 vaccines using an optimization-based strategy, *Process Integration and Optimization for Sustainability*, 5(1), 3-12.
- Pereira D.P., Melicio R., Mendes V.M.F., 2021, A team allocation decision for aircraft fleet maintenance, *IOP Conference Series: Materials Science and Engineering*, 1024(1), 012102.
- Short M., Isafiade A.J., 2020, Network synthesis including detailed packed bed column designs in python MExNetS - An open-source package for mass exchanger, *Chemical Engineering Transactions*, 81, 817-822.
- Sy, C., Aviso, K., Cayamanda, C., Chiu, A., Lucas, R., Promentilla, M., Razon, L., Tan, R., Tapia, J., Torneo, A., Ubando, A., Yu, D., 2020, Process integration for emerging challenges: optimal allocation of antivirals under resource constraints, *Clean Technologies and Environmental Policy*, 22, 1359–1370.
- Torres J.J., Tinjaca C.D., Alvarez O.A., Gomez J.M., Optimization proposal for emulsions formulation considering a multiscale approach, *Chemical Engineering Science*, 212, 115326.