Simulation of Mass Vaccination Programs using GPenSIM in Matlab

A Practical and Experimental approach between centralized and mobile vaccination programs

Mohammed Z. Guniem   
*Department of Electrical Engineering and Computer Science*  
*University of Stavanger*Stavanger, Norway  
m.guniem@stud.uis.no

Abstract

Abstract comes here.

Keywords

Vaccination-Programs, Simulation, Petri-Nets, GPenSIM, Matlab.

Motivation

Vaccination has long been a powerful tool in providing immunity against infectious diseases, which have otherwise been far more deadly without the mass production and distribution of effective vaccines to provide immunity against such deadly diseases.

Figure 1 below shows the fatality rate of major virus outbreaks worldwide in the last 50 years as of January 2020 provided by “statista.com” [1], a clear decrease of the fatality rate from 80% of the Marburg disease in 1967 to 9.6% of the SARS virus disease in 2002 highlights the importance and benefits of vaccination in the fight against new viruses and diseases.

This paper describes a practical project that aims to measure the effectiveness of a traditional centralized vaccination program in comparison with a more mobile vaccination program. The main goal of this project is to utilize the capabilities of the GPenSIM simulation package in MATLAB to establish a scientific proof of the strengths and weaknesses of the mentioned vaccination programs and their potentials in mass vaccination of the human population.

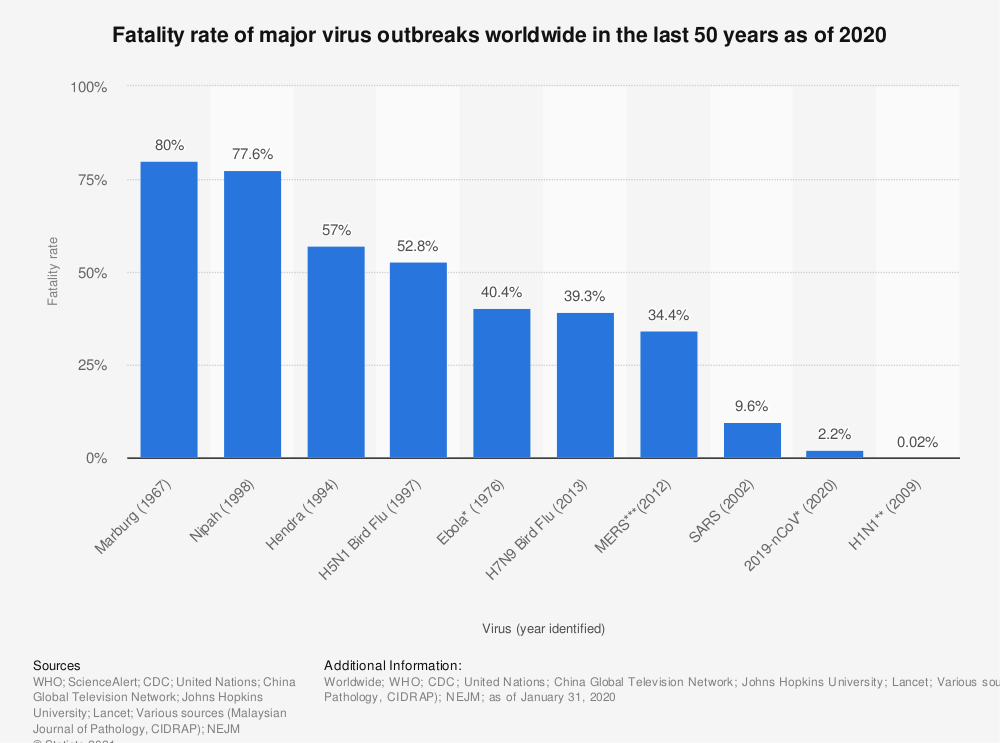


Figure 1, Fatality rate of some major virus outbreaks since 1976, [1].

# **Introduction**

The experienced history of vaccination has proven its effectiveness in protecting both human and animal populations against various diseases since its introduction in 900 CE [2]. However, most of the research has since then been focusing on developing the right vaccine against new outbreaks of diseases, with little effort to investigate how to distribute the developed vaccines across societies in an efficient and agile approach. The speed and effectiveness in which vaccine doses are distributed to society members is an important factor to limit and prevent the outbreaks of diseases across the human population, especially in tight urban environments where individuals are dependent on daily physical contact to keep the society functioning as desired.

This research takes on the challenge of comparing the two main types of vaccination programs; the first type of these programs provides the vaccine doses in a centralized fashion by asking residents to visit a vaccination center, while on the other hand a decentralized vaccination program can be mobile by letting the health crew visit residents at their residential address to provide the vaccine doses.

Each vaccination program has its combination of advantages and disadvantages in term of speed, quality and environmental cost along with some other factors. The centralized vaccination has the advantage of being easy to setup and manage, with the disadvantage of being a contributing factor in spreading infectious diseases as people rush to the vaccination centers and wait in long queues to receive the vaccine, which increases the physical contact of society members and therefore the risk of a higher rate of infections. This physical contact can be minimized when a trained health crew visit residents at their homes to provide the vaccine, but this mobile vaccination program still comes at a cost of more management and coordination of the operations during the vaccination process.

This research will put each vaccination program under the magnifying glass to try revealing their hidden effects and establish a better understanding of their efficiency. We start by building a ground foundation of the process that each individual society member must go through to get a dose of the vaccine in both of the centralized and mobile vaccination programs. Then move on into constructing two petri nets[[1]](#footnote-1) that will be used to implement two separate technical simulations of both vaccination programs using the GPenSIM package tool in MATLAB. The interpretation of these GPenSIM simulations can help achieve the goal of this research by establishing an understanding of the strengths and weaknesses in each vaccination program during various conditions and situations. But at first lets dive into some related work in this field that has been contributed in previous research.

# **Related Work**

Intoduction comes here.

# **Centralized vs. Mobile Vaccination Programs**

The process that each vaccine receiver must go through is described as the workflow of the vaccination program, these workflows can vary in small and big details across environments and up to many factors. In this research we will propose two simplified common approaches of such processes for both centralized and mobile vaccination programs.

Starting with the centralized vaccination program, we assume that visitors arrive at the vaccination center after booking an appointment and that the rate of arrival is known to us, for example 1 visitor per minute. Upon arrival the visitor should be registered at a registration desk before queueing up to receive the vaccine from a trained health personal, then hold in another queue to occupy a waiting room for observation of any unexpected side effects and reactions that might show within half an hour after receiving the vaccine. The process of registering a new visitor is estimated to take 5 minutes on average and requires the help of one staff member at duty. similarly, the vaccination process is expected to take 10 minutes on average and requires the help of one trained health personal to inject the vaccine into the receiver’s body in a safe way. At last, vaccine receivers are required to wait for 30 minutes in isolated waiting rooms available at the vaccination center before leaving. This workflow is visualized below on the left side of figure 2.

The previously proposed workflow of a centralized vaccination program is said to be visitor oriented, as visitors are moving from one stage to another through the vaccination center. This is different for a mobile vaccination program where health personnel are dispatched from one street to another to visit residents at their homes and provide them with the vaccine, making this workflow more oriented to health personal than for the vaccine receivers.

The proposed mobile workflow of this research is provided on the right side of figure 2. It starts by dispatching a vaccination bus with one driver and one health personal to the start street, which is a process that is expected to take 15 minutes on average. After arriving at the target street, the vaccination process can start by visiting residents at their homes in this street, or by letting the residents of the street queue up to receive their dose in the vaccination bus, either way it is estimated to take 10 minutes to provide one dose of vaccine as in the centralized vaccination program. After vaccinating the residents of the targeted street, the vaccination bus needs a turnaround time that is expected to take 1 minute before starting to drive to the next target street.

To keep focus on the main operations of vaccine distribution, we assume that we have an available response team that have sufficient capacity to response to any side effects experienced by any of the vaccine receivers. For this reason, we will not include the details of the intervention from this response team in the main workflows of this research. But it is indeed recommended to extend this research by integrating this response operation into the main workflow as it would make the simulation more realistic and reliable in the future.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Figure 2, The proposed workflows of both centralized(left) and mobile(right) vaccination programs in this research.

# **Simulation Setup of A Centralized Vaccination Program**

Centralized vaccination takes place in a fixed location where residents arrive at the vaccination center to receive a dose of the distributed vaccine. The process of receiving a dose of this vaccine is built around the idea of moving visitors from one stage to another and maintaining several intermediate places or queues between each stage.

According to the proposed workflow on the left of figure 2, centralized vaccination can be divided into 3 stages based on the purpose of each stage:

* Registration

This is the first stage, and it has the purpose of verifying and registering a visitor after arriving at the vaccination center, and before receiving a dose of the vaccine. This stage is expected to take 5 minutes per visitor on average, and it is performed by one staff member per visitor.

* Vaccination

After registration, a visitor moves on to receive a dose of the vaccine from one trained health worker, which is a process that is estimated to take 10 minutes per visitor on average.

* Waiting for any side effects and reactions

Vaccines can cause severe allergic reactions and other side effects that can harm the person receiving the vaccine if it is not dealt with by a team of paramedics available on the premises. Therefore, it is important to wait for at least 30 minutes before leaving the vaccination center. The waiting should also happen in isolated rooms to avoid infections between visitors because the body of the receiver still hasn’t gained the required level of immunity against the targeted disease.

The visitors must stand in queue places waiting for their turn to enter the next stage. Which in turn makes visitors vulnerable to catching the targeted disease while they are at the vaccination center, but the length of these queues can still be controlled by requiring the visitors to book an online appointment in advance before attending to receive the vaccine.

Figure 3 below shows a proposed solution of a petri net solution that visualizes the workflow of this centralized vaccination. In this petri net, we have 4 types of transitions, these transition types are “tVISITOR”, “tREGISTRATION”, “tVACCINATION”, and “tWAITING” each representing the processes of appointment booking, registration, vaccination and waiting respectively.

The number of transitions is dependent on the following factors:

* The number of available resources of staff members to handle the registration of visitors.
* The number of available health workers to handle the vaccination process.
* The number of waiting rooms to be used after receiving the vaccine.

In the petri net drawn in figure 3 there are 1 staff member available who can register 1 visitor per 5 minutes, 2 health workers who can register 2 visitors per 10 minutes, and 6 waiting rooms that can host 6 visitors each 30 minutes. This is the optimal setup of resources as 1 staff member can process 6 visitors in 30 minutes, and 2 health workers can together process 6 visitors in 30 minutes, and none of the less the 6 waiting rooms does also process 6 visitors per 60 minutes. Assigning any more resources at one of these stages can be seen as excessive and unnecessary measure. Also, on the other hand any shortage of resources at any stage can cause ring effects and impact the progress of other stages in the system.

The petri net model realizes the progress from one transition to another in 4 places where tokens in “P1” denote visitors that have arrived at the vaccination center but still has not been registered, while tokens in “P2” denote registered visitors waiting to be vaccinated, and tokens at “P3” denote vaccinated visitors trying to access a waiting room and wait for 30 minutes before leaving the premises. Monitoring the number of tokens at “P4” allow us to track how many visitors the system was able to process at each time of the simulation.

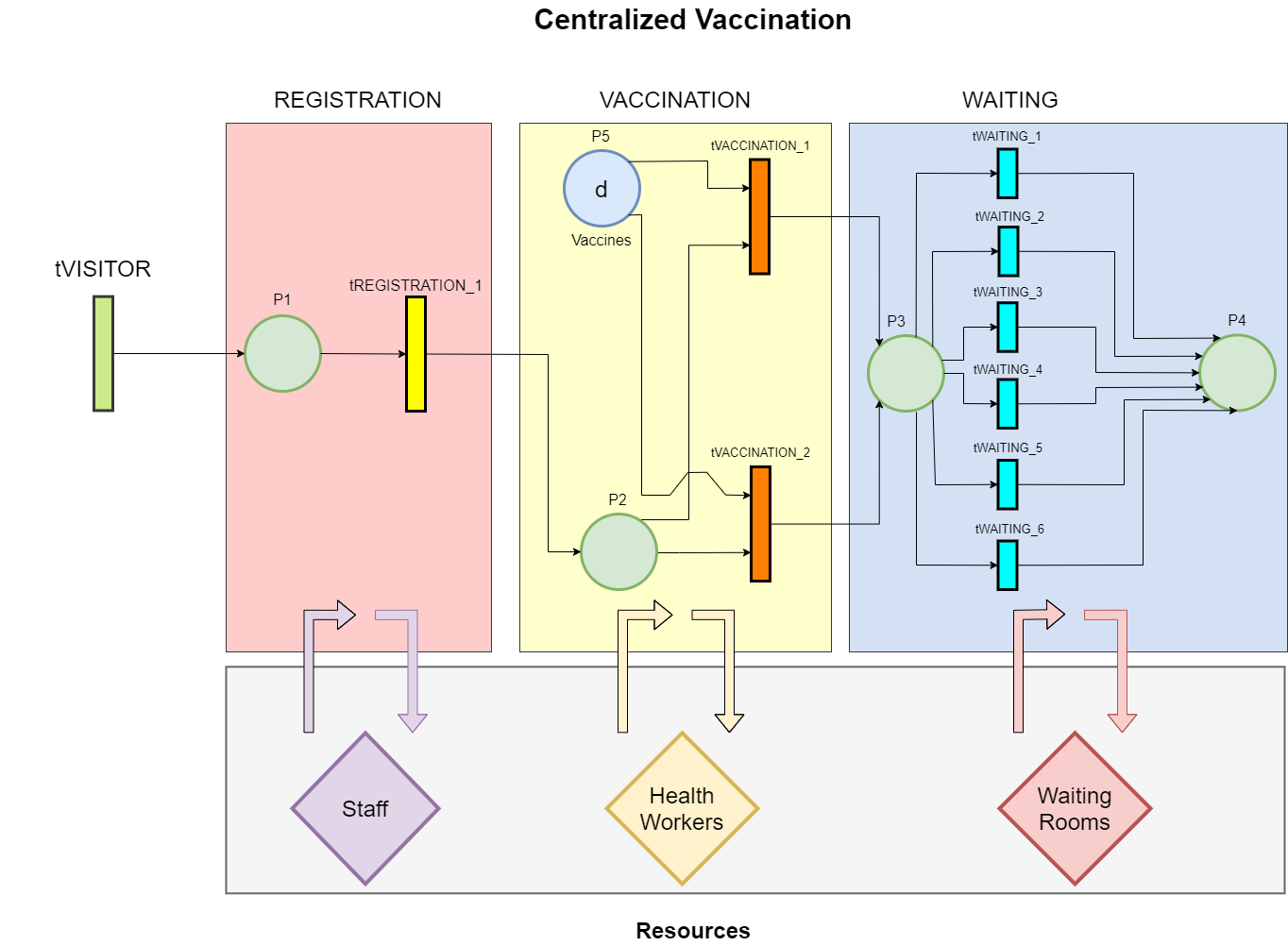


Figure 3, A proposed petri net of centralized vaccination.

# **Implementation of a Centralized Vaccination Program using GPenSIM**

Now that we have constructed a petri net for the centralized vaccination program, it is time to start implementing the logic of this program into a computer simulation using the powerful capabilities of the GPenSIM tools in MATLAB.

The implemented code of this simulation is to be found inside the folder named “Centralized Vaccination” in the attached source code. It consists of the following 4 files:

* Petri net Definition File (PDF)

As its name suggests, this file defines the petri net of the system to be simulated, it contains the 3 necessary sets definitions of the places, transitions and arcs in the petri net shown in figure 3.

* Common Pre-processor File

This file is used by the transitions of the types “tREGISTRATION\_\*”, “tVACCINATION\_\*” and “tWAITING\_\*” to acquire the needed resources before firing, these resources are one staff member, one health worker, and one waiting room, that is needed to enable the firing of the transition types of one “tREGISTRATION\_\*”, one “tVACCINATION\_\*” and one “TWAITING\_\*” transitions respectively.

In addition, this file also implements a pre-processer of the “tVISITOR” transition, that acts as a scheduler and visitor generator to simulate new arriving visitors at the vaccination center, the time interval between visitors, and the number of visitors arriving at each firing is configured as a global variable in the main simulation file, also in this file it is possible to configure a close time of the vaccination center to stop receiving new visitors.

* Common Post-processor File

This file implements the release of any acquired resources after firing from the transition types of “tREGISTRATION\_\*”, “tVACCINATION\_\*” and “TWAITING\_\*”.

* Main Simulation File (MSF)

This file is the one to run in order to kickstart this simulation, it starts by configuring the start and end real time of the simulation, along with the close time where no more visitors arrive at the vaccination center. Then the interval between visitors, and the number of visitors that arrives at each interval is configured for the “tVISITOR” transition.

Furthermore, it is also required to determine the number of available resources from staff members to work with registration of new visitors, health workers to work with the vaccination process and waiting rooms to host visitors for 30 minutes each.

* A Dynamic Configuration File called “construct.m”

To make this simulation easily scalable, a series of methods are implemented in this file to construct the sets of firing times, transitions and arcs in a compatible structure with GPenSIM.

This is because the system contains multiple of the same types of transitions that can fire simultaneously, an example here is when we have an “n” number of staff members, then also an “n” number of “tREGISTRATION\_” transitions are configured to process arriving visitors at “P1” and output tokens at “P2”, the same logic is applied to the “tVACCINATION\_\*” and “tWAITING\_\*” transitions.

The number of vaccine doses in place “P5” is configured by multiplying the number of available waiting rooms by 2 and then by 8 as the simulation is to run for 8 hours and each waiting room can host/process 2 visitors within 1 hour, then we are not going to need more vaccines than the bottleneck capacity of the available waiting rooms. However, and if needed the value of the configured number of available vaccine doses can be manually set by configuring the respective variable called “num\_of\_vaccines” in the main simulation file.

After implementing this simulation in GPenSIM, we are ready to fire it up and note the performance capability of centralized vaccination. In this simulation run, we are going to use the basic set of available resources of 1 staff member, 2 health workers, and 6 waiting rooms. Which gives us a total of 1 “tREGISTRATION\_\*” transition, 2 “tVACCINATION\_\*” transitions, and 6 “tWAITING\_\*” transitions, with their dynamically constructed arcs from and to places 1 to 5 shown in the visualized petri net in figure 3. We will also configure this simulation to receive 1 new visitor per 5 minutes to match the registration capacity of the one and only available staff member, and then set the close time of the “tVISITOR” transition to 15:30 to stop receiving any new visitors half an hour before ending the simulation by processing all visitors who are occupying the waiting rooms for the last time at this simulated working day.

The result of this simulation is shown in figure 4 below where at the final state 86 visitors has been successfully processed by the system and therefore placed in final place “P4”.

It is also good to notice that the system takes about 45 minutes before it starts producing fully vaccinated and approved visitors, this is because any arriving visitor has to spend at least 45 minutes before getting done with the entire vaccination process from the registering process that takes 5 minutes, through vaccination that takes 10 minutes and finally waiting for 30 minutes for any abnormal reactions to the vaccine.

The maximum queue length is 1 vaccinated visitor at “P3” waiting to access a waiting room most of the time in the simulation, while we also experience a maximum waiting number of visitors of 1 person at places “P1” and “P2” in the very early start of starting to receive visitors in the simulation. Which might be caused by the next and previous transitions pre-processing just at the same time. However, a maximum waiting of 1 person in each queue is not concerning since infection control can still be easily maintained if there is no more than one person in line waiting for the next stage.

It is also important to correctly configure the scheduling of arriving visitors according to the capacity of available resources to minimize the number of holding visitors at each intermediate place between transitions, thus preventing infections between holding visitors.

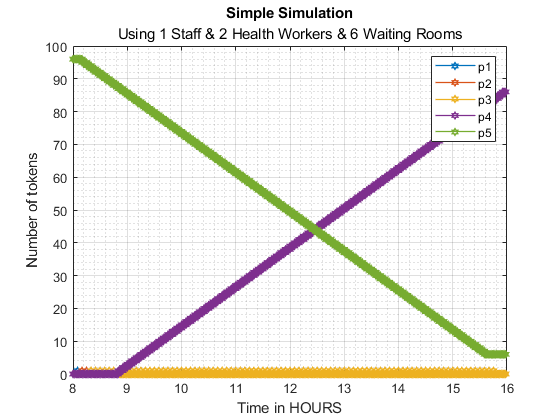


Figure 4, Simulation results of centralized vaccination using GPenSIM, and having a simple set of available resources of 1 staff memebr, 2 health workers, and 6 waiting rooms.

In the upcoming section of study case and scaled performance, we are going to scale this set of available resources to match a real-life situation where we clearly have more resources than what we had for this basic simulation, and then compare it with an equal number of resources used in mobile vaccination to establish an understanding of differences in features and performance between centralized and mobile vaccination on a bigger scale. But first let’s dive into the simulation of mobile vaccination and discuss its features and details.

# **Simulation Setup of A Mobile Vaccination Program**

Mobile vaccination offers residents the opportunity to receive a dose of the distributed vaccine at their residence without having to visit a vaccination center as in centralized vaccination. The idea behind mobile vaccination is to divide the available team of trained health workers in different batches and equip them with a transportation vehicle or what is also known as a vaccination bus. Theses busses are then dispatched to each street in the targeted residence area, and health workers can either move on foot to vaccinate people at their homes or offers the vaccine to the residents of a certain street onboard the vaccination bus.

According to the proposed workflow on the right of figure 2, mobile vaccination can be divided into 3 stages based on the purpose of each stage:

* Dispatch

The operation of mobile vaccination starts with dispatching each health worker to a targeted street using a vaccination bus with its driver. Driving this vaccination bus to a new street address is expected to take 15 minutes on average from one street to another.

* Vaccination

After arriving at a new street, the health worker starts the process of vaccinating street residents one by one. This process is estimated to take 10 minutes on average to vaccinate each resident.

* Completion

This is the turnaround stage that is performed after all residents of the currently visited street have been vaccinated, and the bus can again get dispatched to a new street and so on. The turnaround time is assumed to be 1 minute at this simulation project.

Following this mobile vaccination setup, residents do not have to stand in long queues and can be visited by a health worker to receive a dose of the vaccine. This helps contain the infections as a result of minimized physical contact between residents.

Figure 5 below shows a proposed solution of a petri net that visualizes the workflow of mobile vaccination. In this petri net, we have 3 types of transitions, these transition types are “tDISPATCH\_\*”, “tVACCINATION\_\*” and “tCOMPLETION\_\*” each representing the processes of dispatching, vaccination and turnaround completion respectively.

The number of transitions is dependent on the following factors:

* The number of available bus drivers to reach out to new streets.
* The number of available health workers to handle the vaccination process of street residents upon arrival.

In the petri net shown in figure 4, we have chosen a small setup of 2 available bus drivers and 2 health workers for demonstration purposes, and because all stages do in fact go on concurrently, we must dynamically configure 2 “tDISPATCH\_\*” transitions, 2 “tVACCINATION\_\*” transitions and 2 “tCOMPLETION\_\*” transitions respectively for each stage of this mobile vaccination program, and one for each group of a health worker accompanied with a bus driver.

To keep this simulation as discrete as possible, the arcs between the dispatch transitions and the place “P5” outputs the configured average number of residents in each street of the targeted residence area denoted as “R”, which can be calculated by dividing the total number of residents in an area on the total number of streets in this residence area. This same number of residents “R” is consumed and outputted by the completion transitions upon the end of the vaccination process in a visited street, which is realized in the arcs from the place “P7” to each of the completion transitions, and the arcs from these completion transitions to the place “P8”. The sum of tokens in the places “P7” and “P8” can then be used to measure the number of vaccinated residents at each time of this simulation.

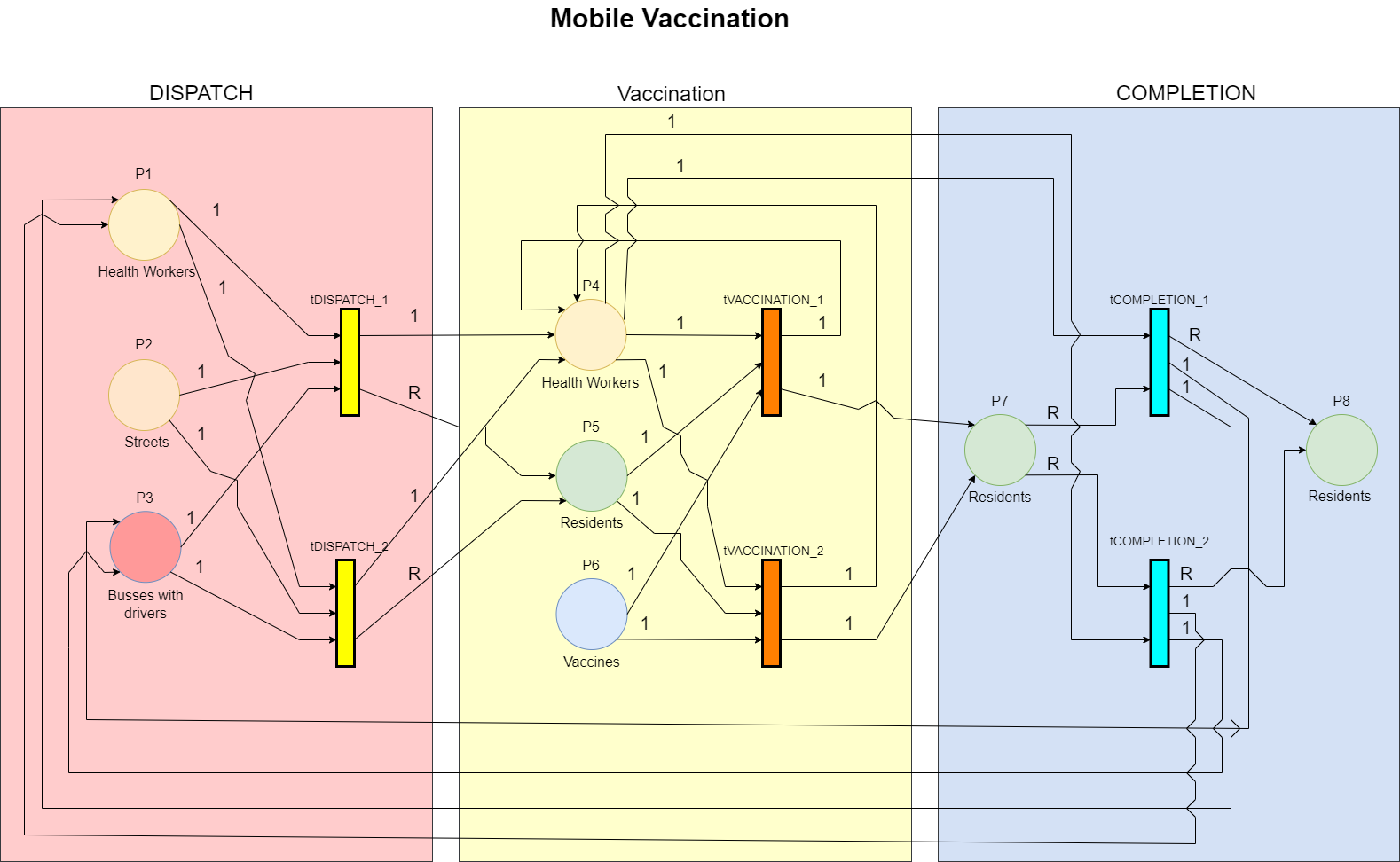


Figure 5, A proposed petri net of mobile vaccination.

# **Implementation of a Mobile Vaccination Program using GPenSIM**

Starting from the petri net of he proposed mobile vaccination program shown in figure 5, we are going to implement its logic into a computer simulation using the available GPenSIM tools in MATLAB.

The technical implementation of this simulation is founded inside the folder named “Mobile Vaccination” in the attached source code of this project; this implementation consists of the following 3 files:

* Petri net Definition File (PDF)

As for the implementation of centralized vaccination, this file also defines the petri net of the mobile vaccination program, it contains the 3 necessary sets definitions of the places, transitions and arcs in the petri net shown in figure 5.

* Main Simulation File (MSF)

This file is the one to run in order to kickstart the simulation of mobile vaccination, it starts by configuring the start and end real time of this simulation. And requires the user to set the number of available health workers and vaccination busses with their drivers to be used under this simulation, along with the number of targeted streets to visit and vaccinate its residents.

Furthermore, the number of available doses of vaccine is set to equal the given total number of residents to vaccinate in the entire residence area. And, an important constant to set is the average number of residents per street in the targeted area, because it is needed to be used as a weight value of the arcs stretched from the “tDISPTACH\_\*” transitions to the place “P5” according to the petri net in figure 5 to simulate the start of the vaccination stage upon arrival at a targeted street. And accordingly, from the place “P7” to the “tCOMPLETION\_\*” transitions to simulate the end of the vaccination stage at a targeted street and turnaround functionality to visit a new street.

* A Dynamic Configuration File called “construct.m”

Just as in the implementation of centralized vaccination and for the purpose of making this simulation easily scalable, a series of methods are implemented in this file to construct the sets of firing times, transitions and arcs in a compatible structure with GPenSIM.

This is because the system contains multiples of the same types of transitions that can fire simultaneously, an example here is when we have an “n” number of health workers and an “n” number of bus drivers, then also an “n” number of “tDISPATCH\_\*” transitions are configured to dispatch a vaccination bus to a targeted street. The same logic is applied to dynamically configure an “n” number of “tVACCINATION\_\*” transitions and an “n” number of “tCOMPLETION\_\*” transitions.

To test our simulation in a simple way, we start by defining a small set of 2 health workers accompanied with 2 bus drivers and 2 vaccination busses, their mission is to vaccinate as much residents as possible from a total of 100 streets, giving an average of 20 residents per street, which is the number “R” shown by the petri net in figure 5. This simulation will then run from 8 am to 16 pm to simulate one working day.

Figure 6 below shows the number of tokens in each place during the simulation, and at first, we are interested on how many residents can get vaccinated by a group of 2 repeatedly dispatched vaccination busses, which is the sum of tokens at the places “P7” and “P8” at 16 O’clock when the simulation ends, which is found to be 86 tokens, meaning that 86 residents have been fully vaccinated during this simulated day of mobile vaccination. This is surprisingly equal to the number of fully vaccinated residents in the implementation of centralized vaccination with similar available resources of health workers performed earlier in this project.

In addition to this observation, it is possible to notice the repeated waves of suddenly increasing and gradually decreasing number of tokens in place “p5”, where each period illustrates the vaccination process that is in progress at the targeted streets.

It is also useful to notice that the final number of streets left to vaccinate is 94 tokens left at place “p2”, meaning that 6 streets has been visited since we started from 100 streets in total. However, this does not mean that all residents in these 6 streets have been vaccinated as we can clearly see that there are still tokens in place “P5” at the end time of this simulation.

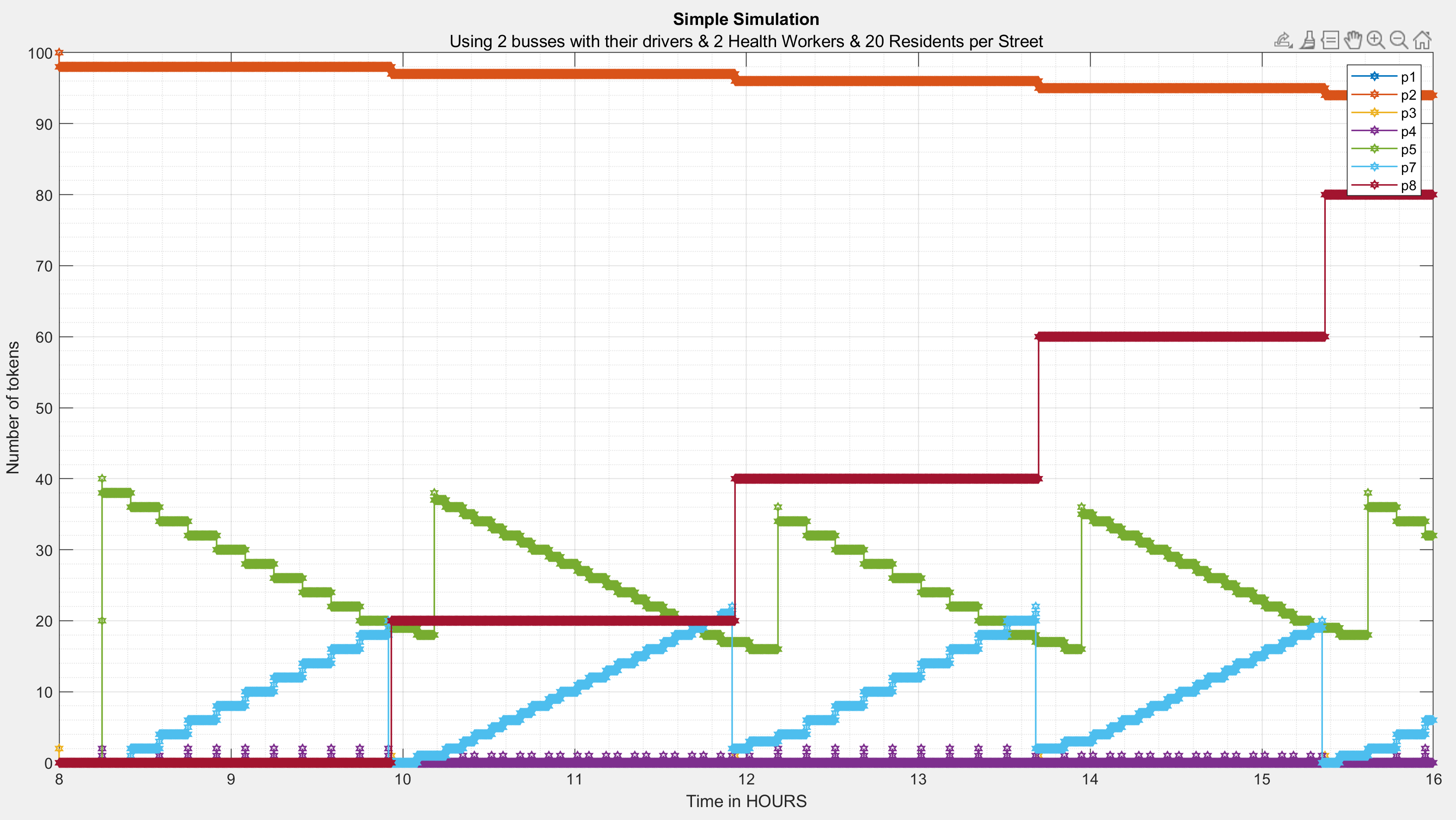


Figure 6, Simulation results of mobile vaccination using GPenSIM, and having a simple set of available resources of 2 health workers accompanied with 2 vaccination busses driven by 2 available drivers.

In the upcoming section of study case and scaled performance, we are going to scale this set of available resources to match a real-life situation where we clearly have more resources than what we had for this basic simulation of mobile vaccination, and then compare it with a similar number of resources used in centralized vaccination to establish an understanding of differences in features and performance between centralized and mobile vaccination on a bigger scale.

# **Study Case and Scaled Performance Comparsion between the centralized and mobile vaccination programs.**

Text comes here.

# **Future Improvements.**

Text comes here.

*Conclusion*

Text comes here.

##### **Acknowledgment**

Remember to thank the teacher here.

##### **References**

References comes here.

[1] Statista, Fatality rate of major virus outbreaks worldwide in the last 50 years as of 2020, 2021. Available: <https://www.statista.com/statistics/1095129/worldwide-fatality-rate-of-major-virus-outbreaks-in-the-last-50-years/>. Accessed on: 06.10.2021.

[2] The College of Physicians of Philadelphia, History of Vaccines, an educational resource by the college of physicians of Philadelphia, 2021. Available: <https://www.historyofvaccines.org/timeline/all>. Accessed on: 06.10.2021.

[3] James Lyle Peterson. 1981. Petri Net Theory and the Modeling of Systems. Prentice Hall PTR, USA. page 7.

[1] remove this

1. A Petri net is composed of four parts: a set of places P, a set of transitions T, an input function I, and an output function O. [3] [↑](#footnote-ref-1)