**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені Ігоря Сікорського»**

**Факультет прикладної математики**

**Кафедра прикладної математики**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_ О. Р. Чертов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 р.

**Дипломний проект**

**освітньо-кваліфікаційного рівня «Спеціаліст»**

зі спеціальності 113 «Прикладна математика»

на тему: Математичне та програмне забезпечення системи регіонального розподілу вакцин під час підйому сезонної захворюваності на грип

Виконав: студент II курсу, групи КМ-71мп

|  |  |  |
| --- | --- | --- |
|  | Алістратенко Микита Олексійович | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Керівник | доцент  Соловйов С.О. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Консультант із нормоконтролю | старший викладач Мальчиков В. В. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Рецензент | канд. техн. наук. доцент,  Дідковська Марина Віталіївна | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Засвідчую, що в цьому дипломному проекті немає запозичень із праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_\_\_\_\_\_\_\_

Київ — 2018**Національний технічний університет України**

**«Київський політехнічний інститут**

**імені Ігоря Сікорського»**

Факультет прикладної математики

Кафедра прикладної математики

Освітньо-кваліфікаційний рівень «Магістр»

Спеціальність 113 «Прикладна математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_ О. Р. Чертов

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 р.

**ЗАВДАННЯ**

**на магістерську дисертацію**

Алістратенко Микити Олексійовича

1. Тема проекту: «Математичне та програмне забезпечення системи регіонального розподілу вакцин під час підйому сезонної захворюваності на грип»,

керівник проекту Соловйов Сергій Олександрович, доцент,

затверджені наказом по університету від «\_\_» \_\_\_\_\_\_\_ 2018 р. № \_\_\_\_\_.

2. Термін подання студентом дисертації: «\_\_» \_\_\_\_\_\_\_ 2018 р.

3. Об’єкт дослідження: диференційні рівняння, диференційні системи; дискретні системи; існуючі моделі регіонального розподілу вакцин, методи та параметри які впливають на оптимальне розподілення вакцин, моделі розповсюдження грипу.

4. Предмет дослідження: концептуальна модель системи регіонального розподілу вакцин.

5. Перелік завдань, які потрібно зробити: виконати огляд літературних джерел за тематикою проекту, провести аналіз математичних методів регіонального розподілу доз вакцин, провести аналіз існуючих систем, обґрунтувати вибір методу, розробити відповідне програмне забезпечення, провести тестування розробленого програмного забезпечення.

6. Орієнтований перелік ілюстративного матеріалу: порівняльна таблиця існуючих систем, порівняльна таблиця математичних методів, ілюстрація роботи розробленого методу.

7. Орієнтований перелік публікацій: .

8. Дата видачі завдання: «\_\_» \_\_\_\_\_\_\_ 2018 р.

Календарний план

| № з/п | Назва етапів виконання дипломного проекту | Термін виконання етапів проекту | Примітка |
| --- | --- | --- | --- |
| 1 | Ґрунтовне ознайомлення з предметною областю | 01.06.2018 |  |
| 2 | Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури | 12.09.2018 |  |
| 3 | Робота над першим розділом магістерської дисертації | 15.10.2018 |  |
| 4 | Проведення наукового дослідження; робота над статтею за результатами наукового дослідження | 25.10.2018 |  |
| 5 | Робота над другим розділом магістерської дисертації | 07.11.2018 |  |
| 6 | Робота над третім розділом магістерської дисертації | 20.11.2018 |  |
| 7 | Робота над четвертим розділом магістерської дисертації | 30.11.2018 |  |
| 8 | Оформлення текстової та графічної частин магістерської дисертації | 05.12.2018 |  |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алістратенко М.О.

Науковий керівник дисертації \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Соловйов С.О.реферат

Пояснювальну записку виконано на 82 аркушах, вона містить 2 додатки та перелік посилань на використані джерела з 6 найменувань. У роботі наведено 2 рисунків та 37 таблиць.

Метою дипломного проекту є вдосконалення математичної моделі, методів та алгоритмів регіонального розподілу вакцин для досягнення балансу мiж оптимальним розподіленням вакцин та максимальною маргінальною вигодою від одної вакцини.

У рамках проекту проведено аналіз існуючих систем регіонального розподілу вакцин. Прогностичний алгоритм було обрано оптимальним. Розроблювано систему вирішено будувати, орієнтуючись на зібраних даних про хворих осіб на грип в кожному регіоні України за 2016 та 2017 роки.

За результатами дипломного проектування розроблено програмну систему на мові програмування Java.

Розроблену систему можна використовувати в вільному доступі будь – якому органу для розрахунку оптимального розподілення вакцин між регіонами.

Ключові слова: диференційні системи, різницеві рівняння, вакцина, грип.

ABSTRACT

The thesis is presented in 82 sheets. It contains 2 appendixes and bibliography of 6 references. Two figures and 37 table are given in the thesis.

The goal of the diploma project is to improve the mathematical model, methods and algorithms of regional distribution of vaccines to achieve a balance between the optimal distribution of vaccines and the maximum marginal benefit of one vaccine.

An analysis of existing regional vaccine distribution systems was conducted within the framework of the project. The prognostic algorithm was chosen as optimal. It is decided to build the developed system, guided by the collected data on sick persons in the influenza in each region of Ukraine for 2016 and 2017 years.

According to the results of the diploma design, a software system was developed in the Java programming language.

The developed system can be used in the free access of anybody to calculate the optimal allocation of vaccines between regions

Key words: differential systems, difference equations, vaccine, influenza.

Зміст

[Перелік умовних позначень, скорочень і термінів 13](#_Toc531914114)

[ВСТУП 14](#_Toc531914115)

[1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ 16](#_Toc531914116)

[1.1 Вакцина та вакцинація 16](#_Toc531914117)

[1.2 Висновки до розділу 19](#_Toc531914118)

[2 ОПИС ІСНУЮЧОЇ МОДЕЛІ 20](#_Toc531914119)

[2.1 Модель без вакцинації 20](#_Toc531914120)

[2.2 Модель з вакцинацією 22](#_Toc531914121)

[2.3 Висновки до розділу 24](#_Toc531914122)

[3 ВХІДНІ ДАНІ 25](#_Toc531914123)

[4 ПАРАМЕТРИЗАЦІЯ МОДЕЛІ 36](#_Toc531914124)

[5 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ 46](#_Toc531914125)

[5.1 Пропорційний алгоритм 46](#_Toc531914126)

[6 ОПТИМІЗАЦІЯ ІСНУЮЧОГО МЕТОДУ 48](#_Toc531914127)

[6.1 Алгоритм критичного періоду 48](#_Toc531914128)

[6.2 Оптимізований пропорційний алгоритм 49](#_Toc531914129)

[6.3 Прогностичний алгоритм 50](#_Toc531914130)

[7 РЕЗУЛЬТАТИ 52](#_Toc531914131)

[8 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ 75](#_Toc531914132)

[9 МАЙБУТНЯ РОБОТА 79](#_Toc531914133)

[ВИСНОВКИ 80](#_Toc531914134)

[Перелік посилань 82](#_Toc531914135)

[ДОДАТОК А Лістинг програми 83](#_Toc531914136)

[ДОДАТОК Б Ілюстрований матеріал 96](#_Toc531914137)

Перелік умовних позначень, скорочень і термінів

SQL — structured query language.

ПЗ – програмне забезпечення

ВСТУП

Вакцинація почала використовуватися для боротьби з руйнівними наслідками пандемій грипу з середини 20-го століття і сьогодні вона залишається високоефективним заходом забезпечення імунітету населення, в тому числі під час сезонного підйому захворюваності [1]. Проте через затримку у створенні, виробництві та забезпеченні вакцинами необхідний глибокий аналіз розподілу наявних вакцин, як тільки вони стануть доступними. Зазвичай виготовлення вакцини триває до шести місяців з моменту виявлення нового вірусу грипу, і лише після цього можливим стає забезпечення вакцинами населення [2]. Як результат, вакцини проти сезонного штаму грипу можуть стати доступними тільки тоді, коли захворюваність населення досягає максимуму або навіть після того, як хвиля захворюваності почне вщухати. Запаси вакцин повинні бути розподілені так, щоб забезпечити їх використання з максимальним ефектом. З огляду на небажання багатьох людей вакцинуватися проти грипу, теоретичні дослідження показують, що відсоток осіб, які потребують вакцинації для забезпечення адекватного захисту всього населення, варіює від 30% до 50% [3].

Ефективне та своєчасне забезпечення вакцинами в умовах обмежених ресурсів є надзвичайно важливим і визначає актуальність даної роботи. Український центр Грипу та інших ГРВІ в структурі Центру Громадського здоров’я постійно проводить моніторинг епідемічну ситуацію щодо грипу та вакцинації населення. Отже, на базі Українського Центру грипу та інших ГРВІ можливо впровадити нові алгоритми забезпечення грипозними вакцинами.

Підхід до забезпечення вакцинами проти грипу різних регіонів пропорційно до кількості населення вперше був використаний в США під час пандемії 2009 року. Ефективність такого методу забезпечення вакцинами значно відрізнялося залежно від регіону [4]**.**

Без урахування географічної динаміки епідемічного процесу грипу, було досліджено значну кількість різних стратегій фармацевтичного забезпечення вакцинопрофілактики грипу. Запропоновані методи відрізняються залежно від встановлення пріоритетів для осіб в групі ризику [5] до осіб, які найбільше сприяють поширенню збудника. Хоча такі роботи є важливими, увага зосереджена на вирішенні задачі вищого рівня для забезпечення вакцинами різних епідеміологічно неоднорідних регіонів на основі інформації щодо епідемічного процесу грипу у кожному з них.

Іншим може бути підхід, який полягає в оптимальному розподілі обмеженої кількості вакцин протягом певного періоду з огляду на динаміку епідемічного процесу вірусних інфекцій.

# ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

## Вакцина та вакцинація

Вакцинація — введення антигенного матеріалу з метою породити імунітет до хвороби, який запобігає зараженню або ослабляє його негативні наслідки. Як антигенний матеріал використовують:

* живі, але ослаблені штами мікроорганізмів або вірусів;
* убиті (неактивовані) мікроби;
* очищений матеріал, наприклад білки мікроорганізмів;
* синтетичні вакцини.

За даними Європейського регіонального бюро Всесвітньої організації охорони здоров'я, планова імунізація проти поліомієліту, правця, дифтерії, кашлю, кору і епідемічного паротиту («свинки») щороку врятовує життя і здоров'я 3 мільйонам дітей у світі. А за допомогою нових вакцин, які буде розроблено в найближчих 5-10 років, можна буде запобігти загибелі ще 8 мільйонів дітей на рік.

Дієвість вакцинації було вперше виявлено народною медициною. З давніх часів в Індії і Китаї практикували інокуляцію — щеплення рідиною з пустул хворих на легку форму натуральної віспи. Недоліком інокуляції було те, що з інокуляційним матеріалом вводився патогенний неослаблений вірус, що інколи призводило до летальних випадків.

У Англії існувала прикмета, що доярки, які перехворіли на коров'ячу віспу (яка не є небезпечним для людини захворюванням), ніколи не захворюють на натуральну віспу, яка свого часу була поширеним смертельним та інвалідизуючим інфекційним захворюванням людей.

Англійський аптекар і хірург Едуард Дженнер вирішив перевірити правильність цієї прикмети своїми дослідженнями, і повністю підтвердив її вірність. 14 травня 1796 року Дженнер прищепив вміст пухирця коров'ячої віспи 8-річному Джеймсу Фіппсу, а через півтора місяці — вміст пухирця від хворого на натуральну віспу — і хлопчик не захворів, тобто Дженнер зробив хлопчикові інокуляцію вірусу коров'ячої віспи. Сам Дженнер переніс інокуляцію вірусу натуральної віспи у 8-річному віці й, маючи імунітет від неї, на собі перевірити правильність своєї гіпотези не міг. Так було експериментально доведено можливість відносно безпечних профілактичних щеплень.

Проте, у той час даний метод мав обмежені можливості, оскільки він базувався на випадковості, що полягає в існуванні в природі двох споріднених хвороб різної сили. Лише сто років по тому французькому мікробіологу Луї Пастеру вдалося цілеспрямовано ослабити патогенність збудників інших захворювань і приготувати з них препарати для щеплень. У 1881 році він створив захисні препарати проти сибірки, а в 1885 році — проти сказу.

Саме Луї Пастер запропонував називати такі препарати вакцинами, а процедуру їхнього використання — вакцинацією (від лат. vacca — корова).

Побічні ефекти й ускладнення у різних вакцин різні. Найчастіше спостерігають слабо виражені реакції: помірне підвищення температури тіла, почервоніння і біль у місці ін'єкції. У дітей часто спостерігають тривалий плач, погіршення апетиту. Як у будь-якого чужорідного матеріалу можливі алергійні реакції, зокрема (рідко) — набряк Квінке, анафілактичний шок. Деякі живі вакцини здатні спричинити реакції, що нагадують легкий перебіг тих захворювань, для профілактики яких їх вводять. Наприклад, вакцинація від кору, краснухи і епідемічного паротиту в 5% випадків спричинює нерясний висип.

В Україні повнолітнім дієздатним громадянам профілактичну вакцинацію проводять за їх згодою після надання об'єктивної інформації про вакцини, наслідки відмови від них та можливі поствакцинальні ускладнення. Особам, які не досягли п'ятнадцятирічного віку чи визнані у встановленому законом порядку недієздатними, вакцинацію проводять за згодою їх об'єктивно поінформованих батьків або інших законних представників. Особам віком від п'ятнадцяти до вісімнадцяти років чи тих, які визнані судом обмежено дієздатними, вакцинацію проводять за їх згодою після надання об'єктивної інформації та за згодою об'єктивно поінформованих батьків або інших законних представників цих осіб. Якщо особа та (або) її законні представники відмовляються від обов'язкової вакцинації, лікар має право взяти у них відповідне письмове підтвердження цього, а в разі відмови дати таке підтвердження, засвідчити це письмовим актом у присутності свідків.

Невдовзі після розробки першої вакцини з'явився і рух антивакцинаторів, які висловлювали сумнів щодо безпеки і ефективності вакцин. Втім, як відзначають експерти ВООЗ, більшість аргументів антивакцинаторів не підтверджуються доказовими науковими даними.

Один з популярних аргументів противників вакцинації — припущення про вміст в деяких вакцинах різних токсичних речовин, зокрема консерванту тіомерсал у на основі ртуті. Доказів системної токсичної дії вакцинних консервантів (при застосуванні у стандартних дозах) немає. Передбачуваний зв'язок тіомерсалу з розвитком аутизму також не знайшов підтвердження.

Батьки часто побоюються випадків неправильного зберігання, транспортування, вживання вакцин.

У 2017 році у Австралії за відмову від вакцинації більше 140 тис. сімей було позбавлено платежів на народження дитини.

Зазвичай під терміном біотероризмом мають на увазі таємне вживання біологічних засобів з метою уражання людей і створення психологічного впливу на населення в цілому. Найпростіший і небезпечний спосіб біотероризму — використання біологічних засобів, що проникають всередину організму через шкіру, легені або шлунково-кишковий тракт (засоби біотероризму можуть знаходитися в повітрі, воді, їжі та на предметах в місцях масового скупчення населення). В сучасному світі головним джерелом біологічної небезпеки служать збудники інфекційних захворювань, які можуть бути використані як засоби біотероризму.

У 1972 році більшість країн світу ратифікували конвенцію про заборону біологічної зброї. Проте, на даний момент неможливо повністю усунути загрозу вживання цього виду зброї окремими країнами, які ведуть секретні роботи з високо вірулентними мікроорганізмами під приводом створення звичайних засобів біологічного захисту. Сучасна історія знає немало випадків викиду вирощених в лабораторних умовах збудників особливо небезпечних інфекцій у довкілля. При цьому найбільшу небезпеку являють мікроорганізми, отримані методами генної інженерії.

## Висновки до розділу

В даному розділі було наведену загальну інформацію про вакцини, методи вакцинації на необхідність вакцинації населення. Завдяки цій інформації стає зрозумілим необхідність створення надійного методу розподілення вакцин для зменшення захворювань на протязі року та мінімізування витрат держави.

# ОПИС ІСНУЮЧОЇ МОДЕЛІ

## Модель без вакцинації

Графічно модель процесу розповсюдження грипу без вакцинації описується двома станами: стан здорової людини и стан хворої. Перехід між станами описується двома швидкостями – направлені стрілки на рисунку 2.1.

У моделі є два стани: здоровий стан (S) та хворий (I). Особа може переходити в один і той самий стан лише 1 раз. Одночасно особа може знаходитися в лише в одному стані. Кількість осіб які переходять зі стану в стан визначається швидкостями (стрілки на рисунку 2.1). Швидкість переходу зі стану S в стан I = , а зі стану I в стан S = .

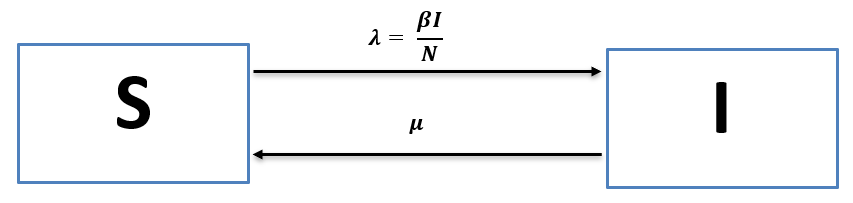


Рисунок 2.1. - Епідемічна модель грипу без вакцинації.

S – кількість здорових осіб,  
I – кількість хворих осіб,  
 – параметр передачі збудника,  
N – загальна кількість осіб в регіоні,  
 – швидкість, з якою особа зі стану I переходять в стан S; так як модель розглядається з проміжком часу в місяць, цей параметр буде дорівнювати 1 – за 1 місяць всі особи переходять зі стану I в стан S.

Для математичного опису динамічних систем використовують систему диференційних рівнянь.

(2.1)

На практиці, використання диференційних рівнянь є складним, тому доцільно перейти від цього представлення до різницевих рівнянь.

(2.2)

Для моделювання епідемічного процесу грипу на основі існуючих даних, потрібно виразити параметр .

(2.3)

Даний параметр є ключовим у прогнозуванні епідемічного процесу. Отриманий параметр оснований на існуючих даних потрібно урівноважити, для наступного використання у генерації кількості хворих та здорових осіб.

## Модель з вакцинацією

Для вирішення поставленої задачі, потрібно у вже описану модель ввести вакцинацію. При введенні вакцинації, вище описана модель приймає вигляд (рисунок 2.2):

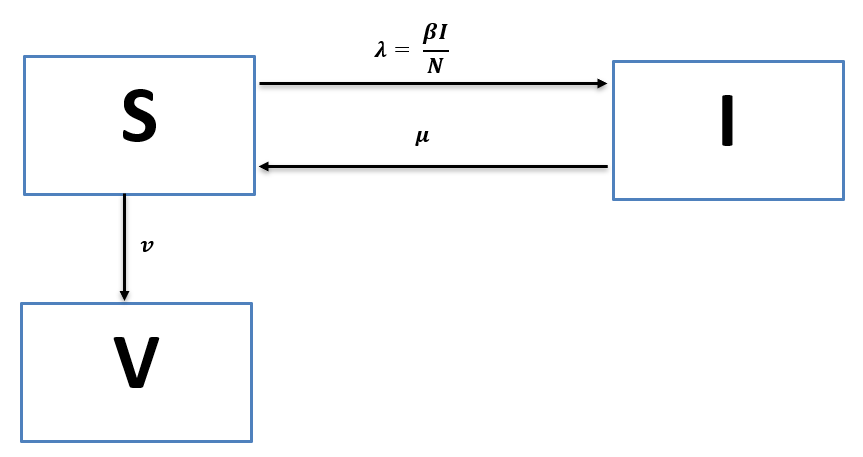


Рисунок 2.2. - Епідемічна модель грипу з вакцинацією.

При включенні вакцинації до моделі, з’являється 3й стан системи – “вакцинований”. В даний стан особа може перейти зі стану S. Швидкість переходу з S в V = . Цей параметр – частина осіб зі стану S яка буде вакцинована.

Данна модель у вигляді диференційних систем матиме вигляд:

(2.4)

Переходячи до різницевих рівнянь для використання у нашій системі, модель матиме вигляд:

(2.5)

Параметр залишається тим же, що і для моделі без вакцинації:

(2.6)

## Висновки до розділу

У цьому розділі описані дві моделі розповсюдження грипу. Обидві системи будуть використовуватись надалі для знаходження оптимального рішення для вирішення поставленої задачі.

# ВХІДНІ ДАНІ

Для представленої системи данні про популяцію по регіонам були отримані з сайту Державної службі статистики України [6]:

* Вінницька обл. – 1590.4 тис.,
* Волинська обл. - 1041 тис.,
* Дніпропетровська обл. – 3230.4 тис.,
* Донецька обл. – 4244 тис.,
* Житомирська обл. – 1240.5 тис.,
* Закарпатська обл. – 1258.8 тис.,
* Запорізька обл. – 1739.5 тис.,
* Ів.-Франківська обл. – 1379.9 тис.,
* Київська обл. – 1734.5 тис.,
* Кіровоградська обл. – 965.8 тис.,
* Луганська обл. – 2386.5 тис.,
* Львівська обл. - 2534 тис.,
* Миколаївська обл. – 1150.1 тис.,
* Одеська обл. – 2386.5 тис.,
* Полтавська обл. – 1426.8 тис.,
* Рівненська обл. – 1162.7 тис.,
* Сумська обл. – 1104.5 тис.,
* Тернопільська обл. – 1059.2 тис.,
* Харківська обл. – 2701.2 тис.,
* Херсонська обл. – 1055.6 тис.,
* Хмельницька обл. – 1285.3 тис.,
* Черкаська обл. – 1231.2 тис.,
* Чернівецька обл. – 908.1 тис.,
* Чернігівська обл. – 1033.4 тис.,
* м. Київ – 2925.8 тис.

Далі, на таблицях 3.1 – 3.10 представленні данні по інтенсивності показників захворюваності на грип за 2016 та 2017 роки по Україні.

Таблиця 3.1 - Інтенсивність показників захворюваності на грип за першу половину 2016 року в перших чотирнадцяти регіонах України.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Вінницька | 64.2 | 15.5 | 1.2 | 0.0 | 0.0 | 0.0 |
| Волинська | 0.5 | 2.0 | 1.1 | 0.0 | 0.0 | 0.0 |
| Дніпропетровська | 24.3 | 56.1 | 3.8 | 0.0 | 0.0 | 0.0 |
| Донецька | 59.1 | 21.6 | 2.7 | 0.1 | 0.0 | 0.0 |
| Житомирська | 155.5 | 164.2 | 12.7 | 0.7 | 0.0 | 0.1 |
| Закарпатська | 1.8 | 4.3 | 0.2 | 0.0 | 0.0 | 0.0 |
| Запорізька | 65.3 | 125.3 | 6.0 | 0.0 | 0.2 | 0.0 |
| Ів.-Франківська | 3.8 | 8.7 | 1.9 | 0.7 | 0.0 | 0.0 |
| Київська | 76.8 | 54.1 | 6.0 | 0.0 | 0.0 | 0.0 |
| Кіровоградська | 198.0 | 115.9 | 4.4 | 0.6 | 0.1 | 0.0 |
| Луганська | 21.4 | 17.7 | 1.4 | 0.0 | 0.0 | 0.0 |
| Львівська | 62.4 | 96.8 | 11.5 | 1.0 | 0.1 | 0.0 |
| Миколаївська | 24.3 | 61.1 | 11.9 | 0.8 | 0.0 | 0.1 |
| Одеська | 157.8 | 83.7 | 7.8 | 0.5 | 0.0 | 0.0 |

Таблиця 3.2 - Інтенсивність показників захворюваності на грип за другу половину 2016 року в перших чотирнадцяти регіонах України.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Вінницька | 0.0 | 0.0 | 0.0 | 0.1 | 0.3 | 25.8 |
| Волинська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.6 |
| Дніпропетровська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 23.3 |
| Донецька | 0.0 | 0.0 | 0.0 | 0.0 | 0.3 | 24.4 |
| Житомирська | 0.0 | 0.0 | 0.0 | 0.2 | 0.8 | 188.0 |
| Закарпатська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.3 |
| Запорізька | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 11.2 |
| Ів.- Франківська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 9.1 |
| Київська | 0.0 | 0.0 | 0.0 | 0.0 | 0.3 | 43.2 |
| Кіровоградська | 0.0 | 0.0 | 0.1 | 0.0 | 0.2 | 88.8 |
| Луганська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 12.5 |
| Львівська | 0.0 | 0.0 | 0.0 | 0.0 | 0.2 | 89.9 |
| Миколаївська | 0.0 | 0.0 | 0.0 | 0.3 | 0.3 | 64.0 |
| Одеська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 51.1 |

Таблиця 3.3 - Інтенсивність показників захворюваності на грип за першу половину 2016 року в інших дванадцяти регіонах України.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Одеська | 157.8 | 83.7 | 7.8 | 0.5 | 0.0 | 0.0 |
| Полтавська | 27.4 | 40.3 | 10.9 | 0.9 | 3.7 | 0.0 |
| Рівненська | 280.6 | 107.5 | 2.8 | 0.5 | 0.0 | 0.0 |
| Сумська | 154.0 | 94.2 | 5.2 | 0.1 | 0.0 | 0.2 |
| Тернопільська | 12.7 | 17.8 | 1.8 | 0.4 | 0.0 | 0.0 |
| Харківська | 7.4 | 9.0 | 5.3 | 1.0 | 0.0 | 0.0 |
| Херсонська | 18.1 | 3.9 | 0.1 | 0.0 | 0.0 | 0.0 |
| Хмельницька | 32.2 | 62.2 | 15.3 | 2.5 | 0.3 | 0.0 |
| Черкаська | 69.8 | 94.1 | 18.7 | 3.6 | 0.5 | 0.0 |
| Чернівецька | 14.4 | 23.0 | 2.5 | 0.0 | 0.0 | 0.0 |
| Чернігівська | 74.7 | 261.9 | 36.3 | 0.2 | 0.0 | 0.0 |
| м. Київ | 62.2 | 33.9 | 2.3 | 0.2 | 0.1 | 0.0 |

Таблиця 3.4 - Інтенсивність показників захворюваності на грип за другу половину 2016 року в інших дванадцяти регіонах України.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Одеська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 51.1 |
| Полтавська | 0.0 | 0.0 | 0.0 | 0.3 | 0.0 | 9.7 |
| Рівненська | 0.0 | 0.0 | 0.1 | 0.8 | 1.1 | 162.8 |
| Сумська | 0.4 | 0.2 | 0.0 | 0.4 | 5.1 | 89.0 |
| Тернопільська | 0.0 | 0.0 | 0.0 | 0.0 | 0.6 | 8.7 |
| Харківська | 0.0 | 0.0 | 0.0 | 0.0 | 0.7 | 2.6 |
| Херсонська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 44.4 |
| Хмельницька | 0.0 | 0.0 | 0.0 | 0.2 | 0.8 | 39.3 |
| Черкаська | 0.1 | 0.0 | 0.0 | 0.0 | 1.0 | 10.7 |
| Чернівецька | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 12.5 |
| Чернігівська | 0.0 | 0.0 | 0.0 | 0.0 | 1.2 | 82.7 |
| м. Київ | 0.0 | 0.0 | 0.0 | 0.1 | 0.7 | 91.7 |

Таблиця 3.5 - Інтенсивність показників захворюваності на грип за першу половину 2017 року в перших чотирнадцяти регіонах України.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Вінницька | 38.3 | 6.8 | 46.1 | 0.0 | 0.0 | 0.0 |
| Волинська | 1.3 | 0.0 | 1.3 | 0.0 | 0.0 | 0.0 |
| Дніпропетровська | 8.8 | 0.8 | 9.6 | 0.0 | 0.0 | 0.0 |
| Донецька | 38.5 | 2.5 | 41.4 | 0.1 | 0.0 | 0.0 |
| Житомирська | 136.7 | 7.3 | 145.2 | 0.3 | 0.0 | 0.1 |
| Закарпатська | 2.2 | 0.0 | 2.3 | 0.0 | 0.0 | 0.0 |
| Запорізька | 177.0 | 20.2 | 197.9 | 0.1 | 0.0 | 0.0 |
| Ів.- Франківська | 14.2 | 0.9 | 15.1 | 0.1 | 0.0 | 0.0 |
| Київська | 31.1 | 0.6 | 31.9 | 0.0 | 0.1 | 0.0 |
| Кіровоградська | 97.7 | 12.1 | 116.7 | 1.0 | 0.0 | 0.0 |
| Луганська | 8.9 | 0.1 | 9.1 | 0.0 | 0.0 | 0.0 |
| Львівська | 74.9 | 4.5 | 80.4 | 0.0 | 0.1 | 0.1 |
| Миколаївська | 73.7 | 5.3 | 79.6 | 0.0 | 0.0 | 0.0 |

Таблиця 3.6 - Інтенсивність показників захворюваності на грип за другу половину 2017 року в перших чотирнадцяти регіонах України.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Вінницька | 0.0 | 0.0 | 0.0 | 0.1 | 0.0 | 0.0 |
| Волинська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Дніпропетровська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.2 |
| Донецька | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 |
| Житомирська | 0.0 | 0.0 | 0.0 | 0.1 | 0.6 | 0.5 |
| Закарпатська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Запорізька | 0.0 | 0.0 | 0.0 | 0.1 | 0.1 | 0.1 |
| Ів.- Франківська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 |
| Київська | 0.0 | 30.7 | 0.0 | 0.0 | 0.0 | 0.0 |
| Кіровоградська | 0.0 | 0.0 | 0.0 | 0.1 | 0.0 | 0.0 |
| Луганська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 |
| Львівська | 0.0 | 28.5 | 0.0 | 0.0 | 0.2 | 0.8 |
| Миколаївська | 0.0 | 0.0 | 0.0 | 0.2 | 0.4 | 0.2 |

Таблиця 3.7 - Інтенсивність показників захворюваності на грип за першу половину 2017 року в інших дванадцяти регіонах України.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Одеська | 48.2 | 18.1 | 77.1 | 2.1 | 0.0 | 0.0 |
| Полтавська | 8.2 | 1.6 | 11.4 | 3.5 | 0.0 | 0.0 |
| Рівненська | 94.4 | 12.7 | 107.7 | 0.3 | 0.0 | 0.0 |
| Сумська | 18.9 | 3.4 | 23.9 | 1.0 | 0.4 | 0.0 |
| Тернопільська | 2.2 | 0.0 | 2.2 | 0.0 | 0.0 | 0.0 |
| Харківська | 0.8 | 0.1 | 1.9 | 0.4 | 0.2 | 0.0 |
| Херсонська | 14.2 | 4.4 | 18.8 | 0.0 | 0.0 | 0.0 |
| Хмельницька | 28.9 | 1.9 | 32.8 | 1.0 | 0.9 | 0.3 |
| Черкаська | 45.0 | 5.1 | 51.8 | 1.4 | 0.0 | 0.0 |
| Чернівецька | 2.8 | 0.1 | 2.9 | 0.0 | 0.0 | 0.0 |
| Чернігівська | 121.8 | 40.0 | 164.9 | 0.4 | 0.0 | 0.0 |
| м. Київ | 19.3 | 1.1 | 20.7 | 0.0 | 0.0 | 0.0 |

Таблиця 3.8 - Інтенсивність показників захворюваності на грип за другу половину 2017 року в інших дванадцяти регіонах України.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Одеська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 3.8 |
| Полтавська | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.2 |
| Рівненська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.6 |
| Сумська | 0.0 | 0.0 | 0.1 | 0.0 | 0.4 | 0.3 |
| Тернопільська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Харківська | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.2 |
| Херсонська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Хмельницька | 0.0 | 0.1 | 0.0 | 0.2 | 1.2 | 2.0 |
| Черкаська | 0.0 | 0.0 | 0.0 | 0.1 | 0.1 | 0.7 |
| Чернівецька | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Чернігівська | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| м. Київ | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.4 |

Таблиця 3.9 - Загальна інтенсивність показників захворюваності на грип за 2016 рік в України.

|  |  |
| --- | --- |
| Найменування регіону | Інтенсивний показник сумарної захворюваності |
|
| Вінницька | 107.1 |
| Волинська | 4.1 |
| Дніпропетровська | 107.4 |
| Донецька | 108.2 |
| Житомирська | 522.2 |
| Закарпатська | 6.5 |
| Запорізька | 208.0 |
| Ів.- Франківська | 24.2 |
| Київська | 180.4 |
| Кіровоградська | 408.1 |
| Луганська | 53.1 |
| Львівська | 262.0 |
| Миколаївська | 162.9 |
| Одеська | 301.0 |
| Полтавська | 93.3 |
| Рівненська | 556.2 |
| Сумська | 348.7 |
| Тернопільська | 42.0 |
| Харківська | 26.0 |
| Херсонська | 66.6 |
| Хмельницька | 152.9 |
| Черкаська | 198.3 |
| Чернівецька | 53.5 |
| Чернігівська | 457.0 |
| м. Київ | 191.2 |

Таблиця 3.10 - Загальна інтенсивність показників захворюваності на грип за 2017 рік в України.

|  |  |
| --- | --- |
| Найменування регіону | Інтенсивний показник сумарної захворюваності |
|
| Вінницька | 91.2 |
| Волинська | 2.7 |
| Дніпропетровська | 19.4 |
| Донецька | 82.7 |
| Житомирська | 290.7 |
| Закарпатська | 4.5 |
| Запорізька | 395.4 |
| Ів.- Франківська | 30.4 |
| Київська | 94.4 |
| Кіровоградська | 227.7 |
| Луганська | 18.3 |
| Львівська | 189.4 |
| Миколаївська | 159.4 |
| Одеська | 149.2 |
| Полтавська | 25.0 |
| Рівненська | 215.6 |
| Сумська | 48.3 |
| Тернопільська | 4.4 |
| Харківська | 3.8 |
| Херсонська | 37.4 |
| Хмельницька | 69.3 |
| Черкаська | 104.1 |
| Чернівецька | 5.7 |
| Чернігівська | 327.1 |
| м. Київ | 41.5 |

# ПАРАМЕТРИЗАЦІЯ МОДЕЛІ

Для дослідження оптимального забезпечення протигрипозними вакцинами необхідно визначити параметри епідеміологічної моделі, які дозволять прогнозувати епідеміологічний процес грипу при відомих початкових даних.

При наявних початкових даних (дані кількості осіб в регіоні, відносна кількість хворих у регіоні в кожен проміжок часу) можливо розрахувати масив розрахованих та вирівняних .

Масив розрахункових зображений на таблицях 4.1 - 4.4.

Таблиця 4.1 - Розрахунковий параметр передачі збудника для перших тринадцяти регіонів України першої половину року.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Вінницька | 0.242 | 0.080 | 0.001 | 1.000 | 1.000 | 1.000 |
| Волинська | 4.200 | 0.524 | 0.001 | 1.000 | 1.000 | 1.000 |
| Дніпропетровська | 2.306 | 0.067 | 0.000 | 1.000 | 1.000 | 1.000 |
| Донецька | 0.365 | 0.123 | 0.044 | 0.009 | 1.000 | 1.000 |
| Житомирська | 1.058 | 0.077 | 0.057 | 0.001 | 79.571 | 0.013 |
| Закарпатська | 2.455 | 0.037 | 0.006 | 1.000 | 1.000 | 1.000 |
| Запорізька | 1.920 | 0.048 | 0.000 | 169.958 | 0.006 | 1.000 |
| Ів.-Франківська | 2.264 | 0.217 | 0.346 | 0.002 | 1.000 | 1.000 |
| Київська | 0.706 | 0.112 | 0.000 | 1.000 | 1.000 | 1.000 |
| Кіровоградська | 0.586 | 0.038 | 0.140 | 0.167 | 0.010 | 1.000 |
| Луганська | 0.825 | 0.077 | 0.033 | 0.022 | 1.000 | 1.000 |
| Львівська | 1.553 | 0.119 | 0.090 | 0.077 | 0.013 | 1.000 |
| Миколаївська | 2.513 | 0.196 | 0.065 | 0.001 | 85.938 | 0.012 |

Таблиця 4.2 - Розрахунковий параметр передачі збудника для інших дванадцяти регіонів України першої половину року.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Одеська | 0.531 | 0.093 | 0.065 | 0.002 | 1.000 | 1.000 |
| Полтавська | 1.471 | 0.270 | 0.083 | 4.077 | 0.000 | 1.000 |
| Рівненська | 0.384 | 0.026 | 0.182 | 0.002 | 1.000 | 1.000 |
| Сумська | 0.612 | 0.055 | 0.017 | 0.112 | 17.837 | 2.000 |
| Тернопільська | 1.397 | 0.100 | 0.211 | 0.003 | 1.000 | 1.000 |
| Харківська | 1.214 | 0.594 | 0.179 | 0.001 | 1.000 | 1.000 |
| Херсонська | 0.218 | 0.024 | 0.011 | 1.000 | 1.000 | 1.000 |
| Хмельницька | 1.931 | 0.247 | 0.166 | 0.121 | 0.003 | 1.000 |
| Черкаська | 1.349 | 0.199 | 0.193 | 0.133 | 0.021 | 8.012 |
| Чернівецька | 1.596 | 0.110 | 0.000 | 1.000 | 1.000 | 1.000 |
| Чернігівська | 3.509 | 0.139 | 0.005 | 0.005 | 1.000 | 1.000 |
| м. Київ | 0.545 | 0.068 | 0.091 | 0.500 | 0.010 | 0.091 |

Таблиця 4.3 - Розрахунковий параметр передачі збудника для перших тринадцяти регіонів України першої половину року.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Вінницька | 1.000 | 1.000 | 124.728 | 2.500 | 82.600 | 1.486 |
| Волинська | 1.000 | 1.000 | 1.000 | 1.000 | 576.818 | 2.338 |
| Дніпропетровська | 1.000 | 1.000 | 1.000 | 1.000 | 23309.611 | 0.379 |
| Донецька | 1.000 | 1.000 | 23.341 | 11.000 | 95.182 | 1.578 |
| Житомирська | 1.000 | 1.000 | 238.714 | 3.333 | 236.302 | 0.728 |
| Закарпатська | 1.000 | 1.000 | 1.000 | 1.000 | 318.285 | 7.004 |
| Запорізька | 1.000 | 1.000 | 1.000 | 1.000 | 11217.260 | 15.778 |
| Ів.-Франківська | 1.000 | 1.000 | 1.000 | 1.000 | 9131.580 | 1.551 |
| Київська | 1.000 | 1.000 | 1.000 | 290.115 | 148.800 | 0.721 |
| Кіровоградська | 1.000 | 102.652 | 0.010 | 205.304 | 432.501 | 1.101 |
| Луганська | 1.000 | 1.000 | 1.000 | 1.000 | 12547.652 | 0.713 |
| Львівська | 1.000 | 1.000 | 39.692 | 6.000 | 377.501 | 0.834 |
| Миколаївська | 1.000 | 1.000 | 343.753 | 1.000 | 186.251 | 1.152 |

Таблиця 4.4 - Розрахунковий параметр передачі збудника для інших дванадцяти регіонів України першої половину року.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Одеська | 1.000 | 1.000 | 1.000 | 1.000 | 51144.849 | 0.942 |
| Полтавська | 1.000 | 1.000 | 346.948 | 0.003 | 9714.552 | 0.842 |
| Рівненська | 1.000 | 86.200 | 9.000 | 1.444 | 145.309 | 0.581 |
| Сумська | 0.500 | 0.056 | 44.593 | 11.400 | 17.510 | 0.212 |
| Тернопільська | 1.000 | 1.000 | 1.000 | 562.490 | 15.500 | 0.250 |
| Харківська | 1.000 | 1.000 | 10.000 | 69.964 | 3.684 | 0.303 |
| Херсонська | 1.000 | 1.000 | 1.000 | 1.000 | 44446.779 | 0.320 |
| Хмельницька | 1.000 | 1.000 | 231.111 | 3.667 | 46.364 | 0.735 |
| Черкаська | 0.125 | 1.000 | 1.000 | 96.139 | 11.083 | 4.221 |
| Чернівецька | 1.000 | 1.000 | 1.000 | 992.392 | 12.556 | 0.222 |
| Чернігівська | 1.000 | 1.000 | 1.000 | 1241.487 | 66.616 | 1.474 |
| м. Київ | 110.000 | 10.000 | 0.639 | 9.500 | 137.369 | 0.210 |

Даний масив надає лише спостережувані дані, тому для того, щоб їх можна було використовувати при прогнозуванні епідемічного процесу, потрібно масив цих даних вирівняти.

Для цього були введені нові вектори – кумулятивна річна частка хворих осіб спостережна та прогнозована. Ці вектори розраховуються по формулі:

(4.1)

де – частка осіб хворих на грип.

Так як масив прогнозованих даних залежить від значень масиву розрахункових , то для знаходження вирівняних , потрібно варіювати розрахункові для того, щоб різниця між раніше введеними векторами була мінімальна.

Для цього був використаний градієнтний спуск – метод знаходження локального мінімуму функції за допомогою руху вздовж градієнта. Результати роботи градієнтного списку зображені на таблицях 4.5, 4.6. На таблицях 4.7 та 4.8 зображена різниця між розрахунковими та вирівняними векторами .

Таблиця 4.5 - Вирівняний параметр передачі збудника для кожного регіону України першої половину року.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Вінницька | 0.242 | 0.081 | 0.001 | 0.985 | 0.985 | 0.985 |
| Волинська | 4.200 | 0.524 | 0.001 | 0.941 | 0.929 | 0.918 |
| Дніпропетровська | 2.306 | 0.067 | 0.000 | 1.000 | 1.000 | 1.000 |
| Донецька | 0.364 | 0.126 | 0.044 | 0.009 | 0.994 | 0.994 |
| Житомирська | 1.058 | 0.077 | 0.057 | 0.001 | 79.647 | 0.013 |
| Закарпатська | 2.454 | 0.037 | 0.006 | 0.965 | 0.962 | 0.957 |
| Запорізька | 1.918 | 0.049 | 0.000 | 163.250 | 0.006 | 0.969 |
| Ів.-Франківська | 2.183 | 0.265 | 0.282 | 0.001 | 0.844 | 0.844 |
| Київська | 0.707 | 0.109 | 0.000 | 1.002 | 1.002 | 1.002 |
| Кіровоградська | 0.586 | 0.038 | 0.139 | 0.167 | 0.010 | 0.999 |
| Луганська | 0.826 | 0.076 | 0.033 | 0.022 | 1.000 | 1.000 |
| Львівська | 1.549 | 0.122 | 0.087 | 0.075 | 0.012 | 0.974 |
| Миколаївська | 2.437 | 0.233 | 0.056 | 0.001 | 74.990 | 0.010 |
| Одеська | 0.527 | 0.102 | 0.068 | 0.002 | 0.983 | 0.983 |
| Полтавська | 1.471 | 0.270 | 0.083 | 4.068 | 0.000 | 1.001 |
| Рівненська | 0.384 | 0.027 | 0.182 | 0.002 | 0.996 | 0.996 |
| Сумська | 0.612 | 0.055 | 0.017 | 0.011 | 180.011 | 1.973 |
| Тернопільська | 1.397 | 0.100 | 0.211 | 0.003 | 1.000 | 1.000 |
| Харківська | 1.214 | 0.594 | 0.175 | 0.010 | 0.923 | 0.954 |
| Херсонська | 0.218 | 0.025 | 0.010 | 0.907 | 0.912 | 0.918 |
| Хмельницька | 1.931 | 0.247 | 0.166 | 0.122 | 0.003 | 1.005 |
| Черкаська | 0.530 | 0.170 | 0.187 | 0.132 | 0.021 | 7.939 |
| Чернівецька | 1.596 | 0.110 | 0.000 | 0.999 | 0.999 | 0.999 |
| Чернігівська | 3.510 | 0.139 | 0.005 | 0.005 | 0.940 | 0.940 |
| м. Київ | 0.545 | 0.068 | 0.091 | 0.500 | 0.010 | 0.091 |

Таблиця 4.6 - Вирівняний параметр передачі збудника для кожного регіону України другої половини року.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Вінницька | 0.985 | 0.985 | 122.932 | 2.502 | 84.182 | 3.037 |
| Волинська | 0.890 | 0.879 | 0.872 | 0.866 | 497.478 | 1.625 |
| Дніпропетровська | 1.000 | 1.000 | 1.000 | 1.000 | 23297.761 | 0.239 |
| Донецька | 0.994 | 0.994 | 23.213 | 10.973 | 97.220 | 2.395 |
| Житомирська | 0.997 | 0.997 | 238.643 | 3.331 | 237.740 | 0.720 |
| Закарпатська | 0.952 | 0.947 | 0.943 | 0.940 | 298.783 | 6.204 |
| Запорізька | 0.966 | 0.963 | 0.963 | 0.960 | 10777.367 | 16.123 |
| Ів.-Франківська | 0.826 | 0.818 | 0.829 | 0.843 | 7682.680 | 3.678 |
| Київська | 1.002 | 1.002 | 1.002 | 290.698 | 149.564 | 0.823 |
| Кіровоградська | 0.999 | 102.597 | 0.010 | 205.351 | 433.213 | 1.592 |
| Луганська | 1.000 | 1.000 | 1.000 | 1.000 | 12549.173 | 0.766 |
| Львівська | 0.974 | 0.974 | 38.644 | 5.837 | 366.309 | 0.804 |
| Миколаївська | 0.879 | 0.879 | 302.176 | 0.871 | 160.721 | 1.803 |
| Одеська | 0.983 | 0.983 | 0.983 | 0.983 | 50263.756 | 1.800 |
| Полтавська | 1.001 | 1.001 | 347.274 | 0.003 | 9691.333 | 0.837 |
| Рівненська | 0.996 | 85.879 | 8.968 | 1.443 | 145.669 | 0.948 |
| Сумська | 0.497 | 0.006 | 442.972 | 11.404 | 17.509 | 0.322 |
| Тернопільська | 1.000 | 1.000 | 1.000 | 562.353 | 15.513 | 0.195 |
| Харківська | 1.005 | 1.051 | 1.081 | 76.583 | 3.755 | 0.435 |
| Херсонська | 0.924 | 0.929 | 0.935 | 0.944 | 42288.523 | 1.337 |
| Хмельницька | 1.005 | 1.005 | 232.343 | 3.651 | 44.797 | 0.479 |
| Черкаська | 0.124 | 0.991 | 0.991 | 95.311 | 11.003 | 0.000 |
| Чернівецька | 1.000 | 1.001 | 1.001 | 993.245 | 12.571 | 0.166 |
| Чернігівська | 0.940 | 0.940 | 0.940 | 1166.387 | 62.600 | 1.389 |
| м. Київ | 110.000 | 10.000 | 0.639 | 9.500 | 137.369 | 0.210 |

Таблиця 4.7 - Різниця розрахункових та вирівняних параметрів передачі збудника для кожного регіону України першої половини року.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Вінницька | 0.000 | 0.000 | 0.000 | 0.015 | 0.015 | 0.015 |
| Волинська | 0.000 | 0.000 | 0.000 | 0.059 | 0.071 | 0.082 |
| Дніпропетровська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Донецька | 0.001 | 0.003 | 0.000 | 0.000 | 0.006 | 0.006 |
| Житомирська | 0.000 | 0.000 | 0.000 | 0.000 | 0.076 | 0.000 |
| Закарпатська | 0.000 | 0.000 | 0.000 | 0.035 | 0.038 | 0.043 |
| Запорізька | 0.002 | 0.001 | 0.000 | 6.709 | 0.000 | 0.031 |
| Ів.-Франківська | 0.082 | 0.049 | 0.064 | 0.000 | 0.156 | 0.156 |
| Київська | 0.001 | 0.002 | 0.000 | 0.002 | 0.002 | 0.002 |
| Кіровоградська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.001 |
| Луганська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Львівська | 0.005 | 0.003 | 0.002 | 0.002 | 0.000 | 0.026 |
| Миколаївська | 0.076 | 0.038 | 0.009 | 0.000 | 10.948 | 0.001 |
| Одеська | 0.005 | 0.009 | 0.003 | 0.000 | 0.017 | 0.017 |
| Полтавська | 0.000 | 0.000 | 0.000 | 0.009 | 0.000 | 0.001 |
| Рівненська | 0.000 | 0.001 | 0.000 | 0.000 | 0.004 | 0.004 |
| Сумська | 0.000 | 0.000 | 0.000 | 0.101 | 162.174 | 0.027 |
| Тернопільська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.000 | 0.000 | 0.004 | 0.008 | 0.077 | 0.046 |
| Херсонська | 0.000 | 0.001 | 0.001 | 0.093 | 0.088 | 0.082 |
| Хмельницька | 0.000 | 0.000 | 0.001 | 0.001 | 0.000 | 0.005 |
| Черкаська | 0.819 | 0.029 | 0.006 | 0.002 | 0.000 | 0.072 |
| Чернівецька | 0.000 | 0.000 | 0.000 | 0.001 | 0.001 | 0.001 |
| Чернігівська | 0.001 | 0.000 | 0.000 | 0.000 | 0.060 | 0.060 |
| м. Київ | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

Таблиця 4.8 - Різниця розрахункових та вирівняних параметрів передачі збудника для кожного регіону України другої половини року.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Вінницька | 0.015 | 0.015 | 1.796 | 0.002 | 1.582 | 1.551 |
| Волинська | 0.110 | 0.121 | 0.128 | 0.134 | 79.340 | 0.713 |
| Дніпропетровська | 0.000 | 0.000 | 0.000 | 0.000 | 11.850 | 0.140 |
| Донецька | 0.006 | 0.006 | 0.128 | 0.027 | 2.038 | 0.817 |
| Житомирська | 0.003 | 0.003 | 0.071 | 0.002 | 1.438 | 0.008 |
| Закарпатська | 0.048 | 0.053 | 0.057 | 0.060 | 19.501 | 0.800 |
| Запорізька | 0.034 | 0.037 | 0.037 | 0.040 | 439.893 | 0.344 |
| Ів.-Франківська | 0.174 | 0.182 | 0.171 | 0.157 | 1448.900 | 2.127 |
| Київська | 0.002 | 0.002 | 0.002 | 0.583 | 0.764 | 0.102 |
| Кіровоградська | 0.001 | 0.055 | 0.000 | 0.047 | 0.712 | 0.491 |
| Луганська | 0.000 | 0.000 | 0.000 | 0.000 | 1.521 | 0.053 |
| Львівська | 0.026 | 0.026 | 1.047 | 0.163 | 11.192 | 0.030 |
| Миколаївська | 0.121 | 0.121 | 41.577 | 0.129 | 25.530 | 0.652 |
| Одеська | 0.017 | 0.017 | 0.017 | 0.017 | 881.093 | 0.857 |
| Полтавська | 0.001 | 0.001 | 0.326 | 0.000 | 23.219 | 0.005 |
| Рівненська | 0.004 | 0.322 | 0.032 | 0.002 | 0.359 | 0.367 |
| Сумська | 0.003 | 0.050 | 398.379 | 0.004 | 0.000 | 0.110 |
| Тернопільська | 0.000 | 0.000 | 0.000 | 0.138 | 0.013 | 0.055 |
| Харківська | 0.005 | 0.051 | 8.919 | 6.618 | 0.071 | 0.131 |
| Херсонська | 0.076 | 0.071 | 0.065 | 0.056 | 2158.256 | 1.017 |
| Хмельницька | 0.005 | 0.005 | 1.232 | 0.015 | 1.567 | 0.255 |
| Черкаська | 0.001 | 0.009 | 0.009 | 0.828 | 0.080 | 4.221 |
| Чернівецька | 0.000 | 0.001 | 0.001 | 0.853 | 0.015 | 0.055 |
| Чернігівська | 0.060 | 0.060 | 0.060 | 75.101 | 4.016 | 0.085 |
| м. Київ | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

# ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

## Пропорційний алгоритм

Єдиний метод для визначення кількості вакцинації на регіон який використовується – пропорційний алгоритм. Його суть полягає в тому, що кількість виділених вакцин залежить лише від кількості осіб у регіоні.

Весь доступний запас вакцин в певний проміжок часу пропорційно ділиться між всіма регіонами. А час вакцинації залежить лише від того, коли дози вакцин стають доступними.

Тобто при підвищенні сезонної захворюваності на грип починається створення вакцин, і в середньому після 60 днів вакцинація стає доступна у регіонах.

(5.1)

де – кількість осіб у регіоні, N – загальна кількість осіб в усіх регіонах системи.

Мінусами даного алгоритму є відсутність врахування епідемічного процесу грипу в кожному регіоні.

Даний алгоритм матиме таку логіку:

Крок 1. Введення початкових даних: масив параметрів передачі збудника , кількість осіб в регіоні, початкова кількість хворих осіб, крок для зміни кількості вакцинації.

Крок 2. Генерація кількості хворих, здорових та вакцинованих осіб з початковими даними (кількість вакцинованих = 0, час вакцинації = 0).

Крок 3. Розрахунок загальної кількості осіб у системі.

Крок 4. Розрахунок кількості доз вакцин на кожен регіон в залежності від кількості осіб в ньому.

# ОПТИМІЗАЦІЯ ІСНУЮЧОГО МЕТОДУ

## Алгоритм критичного періоду

Так як існуючий алгоритм розподілу вакцин один та є дуже простим, існує багато рішень по його оптимізації.

Один із варіантів – виділяти вакцини лише тим регіонам, які ще не досягли піку захворюваності. Тобто, виділяти вакцини тим регіонам, у яких крива захворюваності збільшується в даний момент часу, та розподіляти вакцини пропорційно між цими регіонами:

(6.1)

де

Цей підхід "все або нічого" є проблематичним. Регіони, в яких пік епідеміологічних кривих минув, можуть дуже виграти від додаткової вакцинації.

Більш того, невеликі коливання спостережуваних епідеміологічних кривих внаслідок непередбачуваного характеру епідемії можуть неправильно характеризувати регіони, які ще не досягли піку захворюваності. Такі помилки можуть багато коштувати для системи охорони здоров’я.

Алгоритм буде мати таку логіку:

Крок 1. Введення початкових даних: масив параметрів передачі збудника , кількість осіб в регіоні, початкова кількість хворих осіб, крок для зміни кількості вакцинації.

Крок 2. Генерація кількості хворих, здорових та вакцинованих осіб з початковими даними (кількість вакцинованих = 0, час вакцинації = 0).

Крок 3. Розрахунок кривої захворюваності.

Крок 4. Розрахунок кількості доз вакцин на основі кривої захворюваності для кожного регіону в один момент часу.

## Оптимізований пропорційний алгоритм

З огляду на попередній алгоритм, зрозуміло, що можливо вивести більш оптимізовану формулу, по якій розподіляти вакцини.

Як критерій ефективності фармацевтичного забезпечення вакцинопрофілактики можливо використати критерій “граничної вигоди”, який математично описується наступним чином:

(6.2)

де, I – різниця між кількістю хворих осіб в регіоні з та без вакцинації, – кількість виділених вакцин.

А тепер застосуємо даний параметр для оптимізації пропорційного алгоритму. Алгоритм буде мати таку логіку:

Крок 1. Введення початкових даних: масив параметрів передачі збудника , кількість осіб в регіоні, початкова кількість хворих осіб, крок для зміни кількості вакцинації.

Крок 2. Генерація кількості хворих, здорових та вакцинованих осіб з початковими даними (кількість вакцинованих = 0, час вакцинації = 0).

Крок 3. Розрахунок кількості доз вакцин на кожен регіон на основі пропорційного алгоритму.

Крок 4. Вакцинація регіону у кожен проміжок часу.

Крок 5. Вибір даних при яких гранична вигода максимальна.

## Прогностичний алгоритм

Попередній алгоритм показав, що оптимізація регіонального розподілення вакцин по часу покращує оптимальне рішення. Але ще можливо оптимізувати і розподілення самої кількості вакцин на регіон.

Алгоритм буде мати таку логіку:

Крок 1. Введення початкових даних: масив параметрів передачі збудника , кількість осіб в регіоні, початкова кількість хворих осіб, крок для зміни кількості вакцинації.

Крок 2. Генерація кількості хворих, здорових та вакцинованих осіб з початковими даними (кількість вакцинованих = 0, час вакцинації = 0).

Крок 3. Зміна кількості виділених вакцин на регіон та зміна часу вакцинації.

Крок 4. Розрахунок граничної вигоди для кожного набору даних.

Крок 5. Вибір даних при яких гранична вигода максимальна.

Математично результат даного алгоритму буде мати вигляд:

(6.3),

де – кількість хворих в час без вакцинації, – кількість хворих в час з вакцинацією, – кількість виділених вакцин на регіон в час .

Даний алгоритм може розглядати як одну вакцинацію за весь період, так і декілька вакцинацій за певний період. Такий алгоритм матиме таку логіку:

Крок 1. Введення початкових даних: масив параметрів передачі збудника , кількість осіб в регіоні, початкова кількість хворих осіб, крок для зміни кількості вакцинації, кількість вакцинації за період.

Крок 2. Генерація кількості хворих, здорових та вакцинованих осіб з початковими даними (кількість вакцинованих = 0, час вакцинації = 0, кількість вакцинації за період = 1).

Крок 3. Зміна кількості виділених вакцин на регіон, зміна часу вакцинації, зміна кількості вакцинації за період.

Крок 4. Розрахунок граничної вигоди для кожного набору даних.

Крок 5. Визначення умов, за яких гранична вигода максимальна.

# РЕЗУЛЬТАТИ

Далі продемонстровані результати роботи 4-х раніше описаних алгоритмів регіонального розподілу вакцин.

На таблицях 7.1, 7.2 зображені результати регіонального розподілу вакцин на основі існуючого (пропорційного) алгоритму. Так як вакцинація проводиться в жовтні, то і в системі регіонального розподілення вакцин на основі пропорційного алгоритму час вакцинації визначений жовтнем місяцем.

На таблицях 7.3, 7.4 зображені результати регіонального розподілу вакцин на основі алгоритму критичного періоду. Як бачимо, в деяких регіонах, а саме в Луганській, Київській та Сумській областях вакцинація була проведена в травні або червні місяці. Це зумовлено тим, що дані на основі котрих будувалася модель процесу розповсюдження грипу являються на точними, та мають похибку. Похибка може бути зумовлена відсутністю даних у регіоні по деяким місяцям, та наприклад, більшою кількістю задокументованих звернень до локальних медичних пунктів з приводу захворювань на грип.

На таблицях 7.5, 7.6 зображені результати регіонального розподілу вакцин на основі оптимізованого пропорційного алгоритму з оптимізацією по часу.

На таблицях 7.7, 7.8 зображені результати регіонального розподілу вакцин на основі прогностичного алгоритму з кроком 0.1 та кількістю вакцинацій яка дорівнює одному.

На таблицях 7.9, 7.10 зображені результати регіонального розподілу вакцин на основі прогностичного алгоритму з кроком 0.1 та кількістю вакцинацій яка дорівнює двом.

На таблицях 7.11, 7.12 зображені результати регіонального розподілу вакцин на основі прогностичного алгоритму з кроком 0.0001 та кількістю вакцинацій яка дорівнює одному.

На таблицях 7.13, 71.4 зображені результати регіонального розподілу вакцин на основі прогностичного алгоритму з кроком 0.0001 та кількістю вакцинацій яка дорівнює двом.

Таблиця 7.1 - Регіональне розподілення вакцин по регіонам України першої половини року на основі пропорційного алгоритму.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Вінницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Волинська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Дніпропетровська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Донецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Житомирська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Закарпатська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Запорізька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Ів.-Франківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Київська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Кіровоградська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Луганська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Львівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Миколаївська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Одеська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Полтавська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Рівненська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Сумська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Тернопільська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Херсонська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Хмельницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Черкаська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернівецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернігівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| м. Київ | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

Таблиця 7.2 - Регіональне розподілення вакцин по регіонам України другої половини року на основі пропорційного алгоритму.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Вінницька | 0.000 | 0.000 | 0.000 | 0.037 | 0.000 | 0.000 |
| Волинська | 0.000 | 0.000 | 0.000 | 0.024 | 0.000 | 0.000 |
| Дніпропетровська | 0.000 | 0.000 | 0.000 | 0.076 | 0.000 | 0.000 |
| Донецька | 0.000 | 0.000 | 0.000 | 0.099 | 0.000 | 0.000 |
| Житомирська | 0.000 | 0.000 | 0.000 | 0.029 | 0.000 | 0.000 |
| Закарпатська | 0.000 | 0.000 | 0.000 | 0.029 | 0.000 | 0.000 |
| Запорізька | 0.000 | 0.000 | 0.000 | 0.041 | 0.000 | 0.000 |
| Ів.-Франківська | 0.000 | 0.000 | 0.000 | 0.032 | 0.000 | 0.000 |
| Київська | 0.000 | 0.000 | 0.000 | 0.041 | 0.000 | 0.000 |
| Кіровоградська | 0.000 | 0.000 | 0.000 | 0.023 | 0.000 | 0.000 |
| Луганська | 0.000 | 0.000 | 0.000 | 0.056 | 0.000 | 0.000 |
| Львівська | 0.000 | 0.000 | 0.000 | 0.059 | 0.000 | 0.000 |
| Миколаївська | 0.000 | 0.000 | 0.000 | 0.027 | 0.000 | 0.000 |
| Одеська | 0.000 | 0.000 | 0.000 | 0.056 | 0.000 | 0.000 |
| Полтавська | 0.000 | 0.000 | 0.000 | 0.033 | 0.000 | 0.000 |
| Рівненська | 0.000 | 0.000 | 0.000 | 0.027 | 0.000 | 0.000 |
| Сумська | 0.000 | 0.000 | 0.000 | 0.026 | 0.000 | 0.000 |
| Тернопільська | 0.000 | 0.000 | 0.000 | 0.025 | 0.000 | 0.000 |
| Харківська | 0.000 | 0.000 | 0.000 | 0.063 | 0.000 | 0.000 |
| Херсонська | 0.000 | 0.000 | 0.000 | 0.025 | 0.000 | 0.000 |
| Хмельницька | 0.000 | 0.000 | 0.000 | 0.030 | 0.000 | 0.000 |
| Черкаська | 0.000 | 0.000 | 0.000 | 0.029 | 0.000 | 0.000 |
| Чернівецька | 0.000 | 0.000 | 0.000 | 0.021 | 0.000 | 0.000 |
| Чернігівська | 0.000 | 0.000 | 0.000 | 0.024 | 0.000 | 0.000 |
| м. Київ | 0.000 | 0.000 | 0.000 | 0.068 | 0.000 | 0.000 |

Таблиця 7.3 - Регіональне розподілення вакцин по регіонам України першої половини року на основі алгоритму критичного періоду.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Вінницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Волинська | 0.000 | 0.024 | 0.000 | 0.000 | 0.000 | 0.000 |
| Дніпропетровська | 0.000 | 0.076 | 0.000 | 0.000 | 0.000 | 0.000 |
| Донецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Житомирська | 0.000 | 0.029 | 0.000 | 0.000 | 0.000 | 0.000 |
| Закарпатська | 0.000 | 0.029 | 0.000 | 0.000 | 0.000 | 0.000 |
| Запорізька | 0.000 | 0.041 | 0.000 | 0.000 | 0.000 | 0.000 |
| Ів.-Франківська | 0.000 | 0.032 | 0.000 | 0.000 | 0.000 | 0.000 |
| Київська | 0.000 | 0.000 | 0.000 | 0.000 | 0.041 | 0.000 |
| Кіровоградська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Луганська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.056 |
| Львівська | 0.000 | 0.059 | 0.000 | 0.000 | 0.000 | 0.000 |
| Миколаївська | 0.000 | 0.027 | 0.000 | 0.000 | 0.000 | 0.000 |
| Одеська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Полтавська | 0.000 | 0.033 | 0.000 | 0.000 | 0.000 | 0.000 |
| Рівненська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Сумська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.026 |
| Тернопільська | 0.000 | 0.025 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.000 | 0.063 | 0.000 | 0.000 | 0.000 | 0.000 |
| Херсонська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Хмельницька | 0.000 | 0.030 | 0.000 | 0.000 | 0.000 | 0.000 |
| Черкаська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернівецька | 0.000 | 0.021 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернігівська | 0.000 | 0.024 | 0.000 | 0.000 | 0.000 | 0.000 |
| м. Київ | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

Таблиця 7.4 - Регіональне розподілення вакцин по регіонам України другої половини року на основі алгоритму критичного періоду.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Вінницька | 0.000 | 0.000 | 0.000 | 0.037 | 0.000 | 0.000 |
| Волинська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Дніпропетровська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Донецька | 0.000 | 0.000 | 0.000 | 0.099 | 0.000 | 0.000 |
| Житомирська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Закарпатська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Запорізька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Ів.-Франківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Київська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Кіровоградська | 0.000 | 0.000 | 0.023 | 0.000 | 0.000 | 0.000 |
| Луганська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Львівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Миколаївська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Одеська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.056 |
| Полтавська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Рівненська | 0.000 | 0.000 | 0.027 | 0.000 | 0.000 | 0.000 |
| Сумська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Тернопільська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Херсонська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.025 |
| Хмельницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Черкаська | 0.029 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернівецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернігівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| м. Київ | 0.000 | 0.068 | 0.000 | 0.000 | 0.000 | 0.000 |

Таблиця 7.5 - Регіональне розподілення вакцин по регіонам України першої половини року на основі пропорційного алгоритму з оптимізацією по часу.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Вінницька | 0.000 | 0.000 | 0.000 | 0.037 | 0.000 | 0.000 |
| Волинська | 0.024 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Дніпропетровська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.076 |
| Донецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Житомирська | 0.029 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Закарпатська | 0.029 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Запорізька | 0.041 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Ів.-Франківська | 0.032 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Київська | 0.000 | 0.000 | 0.000 | 0.041 | 0.000 | 0.000 |
| Кіровоградська | 0.023 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Луганська | 0.000 | 0.000 | 0.000 | 0.000 | 0.056 | 0.000 |
| Львівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.059 |
| Миколаївська | 0.027 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Одеська | 0.000 | 0.000 | 0.000 | 0.000 | 0.056 | 0.000 |
| Полтавська | 0.033 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Рівненська | 0.000 | 0.000 | 0.027 | 0.000 | 0.000 | 0.000 |
| Сумська | 0.026 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Тернопільська | 0.025 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.063 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Херсонська | 0.000 | 0.025 | 0.000 | 0.000 | 0.000 | 0.000 |
| Хмельницька | 0.030 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Черкаська | 0.029 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернівецька | 0.021 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернігівська | 0.024 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| м. Київ | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.068 |

Таблиця 7.6 - Регіональне розподілення вакцин по регіонам України другої половини року на основі пропорційного алгоритму з оптимізацією по часу.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Вінницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Волинська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Дніпропетровська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Донецька | 0.099 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Житомирська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Закарпатська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Запорізька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Ів.-Франківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Київська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Кіровоградська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Луганська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Львівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Миколаївська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Одеська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Полтавська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Рівненська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Сумська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Тернопільська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Херсонська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Хмельницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Черкаська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернівецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернігівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| м. Київ | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

Таблиця 7.7 - Регіональне розподілення вакцин по регіонам України першої половини року на основі прогностичного алгоритму з кроком 0.1 та кількістю вакцинацій яка дорівнює одному.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Вінницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Волинська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Дніпропетровська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Донецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Житомирська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Закарпатська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Запорізька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Ів.-Франківська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Київська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Кіровоградська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Луганська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Львівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Миколаївська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Одеська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Полтавська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Рівненська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Сумська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Тернопільська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Херсонська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Хмельницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Черкаська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернівецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернігівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| м. Київ | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

Таблиця 7.8 - Регіональне розподілення вакцин по регіонам України України другої половини року на основі прогностичного алгоритму з кроком 0.1 та кількістю вакцинацій яка дорівнює одному.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Вінницька | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Волинська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Дніпропетровська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Донецька | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Житомирська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Закарпатська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Запорізька | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Ів.-Франківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Київська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Кіровоградська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Луганська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Львівська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Миколаївська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Одеська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Полтавська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Рівненська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Сумська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Тернопільська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Херсонська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Хмельницька | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Черкаська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернівецька | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернігівська | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| м. Київ | 0.100 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

Таблиця 7.9 - Регіональне розподілення вакцин по регіонам України першої половини року на основі прогностичного алгоритму з кроком 0.1 та кількістю вакцинацій яка дорівнює двом.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Вінницька | 0.000 | 0.000 | 0.000 | 0.037 | 0.000 | 0.000 |
| Волинська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Дніпропетровська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Донецька | 0.000 | 0.000 | 0.000 | 0.099 | 0.000 | 0.000 |
| Житомирська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Закарпатська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Запорізька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Ів.-Франківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Київська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Кіровоградська | 0.000 | 0.000 | 0.023 | 0.000 | 0.000 | 0.000 |
| Луганська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Львівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Миколаївська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Одеська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.056 |
| Полтавська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Рівненська | 0.000 | 0.000 | 0.027 | 0.000 | 0.000 | 0.000 |
| Сумська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Тернопільська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Херсонська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.025 |
| Хмельницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Черкаська | 0.029 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернівецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернігівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| м. Київ | 0.000 | 0.068 | 0.000 | 0.000 | 0.000 | 0.000 |

Таблиця 7.10 - Регіональне розподілення вакцин по регіонам України другої половини року на основі прогностичного алгоритму з кроком 0.1 та кількістю вакцинацій яка дорівнює двом.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Вінницька | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Волинська | 0.1000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Дніпропетровська | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Донецька | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Житомирська | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Закарпатська | 0.1000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Запорізька | 0.1000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Ів.-Франківська | 0.1000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Київська | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Кіровоградська | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Луганська | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Львівська | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Миколаївська | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Одеська | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Полтавська | 0.1000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Рівненська | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Сумська | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Тернопільська | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Харківська | 0.1000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Херсонська | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Хмельницька | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Черкаська | 0.1000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Чернівецька | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Чернігівська | 0.1000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| м. Київ | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

Таблиця 7.11 - Регіональне розподілення вакцин по регіонам України першої половини року на основі прогностичного алгоритму з кроком 0.0001 та кількістю вакцинацій яка дорівнює одному.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Вінницька | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Волинська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Дніпропетровська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Донецька | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Житомирська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Закарпатська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Запорізька | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Ів.-Франківська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Київська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Кіровоградська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Луганська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Львівська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Миколаївська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Одеська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Полтавська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Рівненська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Сумська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Тернопільська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Херсонська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Хмельницька | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Черкаська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернівецька | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернігівська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| м. Київ | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

Таблиця 7.12 - Регіональне розподілення вакцин по регіонам України другої половини року на основі прогностичного алгоритму з кроком 0.0001 та кількістю вакцинацій яка дорівнює одному.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Вінницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Волинська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Дніпропетровська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Донецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Житомирська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Закарпатська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Запорізька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Ів.-Франківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Київська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Кіровоградська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Луганська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Львівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Миколаївська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Одеська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Полтавська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Рівненська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Сумська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Тернопільська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Херсонська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Хмельницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Черкаська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернівецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернігівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| м. Київ | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

Таблиця 7.13 - Регіональне розподілення вакцин по регіонам України першої половини року на основі прогностичного алгоритму з кроком 0.0001 та кількістю вакцинацій яка дорівнює двом.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| січень | лютий | березень | квітень | травень | червень |
| Вінницька | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Волинська | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Дніпропетровська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Донецька | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Житомирська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Закарпатська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Запорізька | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Ів.-Франківська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Київська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Кіровоградська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Луганська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Львівська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Миколаївська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Одеська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Полтавська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Рівненська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Сумська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Тернопільська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Херсонська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Хмельницька | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Черкаська | 0.0001 | 0.0000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернівецька | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернігівська | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |
| м. Київ | 0.0001 | 0.0001 | 0.000 | 0.000 | 0.000 | 0.000 |

Таблиця 7.14 - Регіональне розподілення вакцин по регіонам України другої половини року на основі прогностичного алгоритму з кроком 0.0001 та кількістю вакцинацій яка дорівнює двом.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Найменування регіону | Місяці | | | | | |
| липень | серпень | вересень | жовтень | листопад | грудень |
| Вінницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Волинська | 0.000 | 0.000 | 0.000 | 0.000 | 0.0001 | 0.000 |
| Дніпропетровська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Донецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Житомирська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Закарпатська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Запорізька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Ів.-Франківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Київська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Кіровоградська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Луганська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Львівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Миколаївська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Одеська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Полтавська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Рівненська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Сумська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Тернопільська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Харківська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Херсонська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Хмельницька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Черкаська | 0.000 | 0.000 | 0.000 | 0.000 | 0.0001 | 0.000 |
| Чернівецька | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| Чернігівська | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| м. Київ | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

Далі на таблицях 7.15 – 7.19 зображені граничні вигоди для всіх вище описаних алгоритмів.

Я к і слід було очікувати, пропорційний алгоритм має найгірші результати. По кожному регіону гранична вигода менша ніж результат по тому ж самому регіону з будь-яким іншим алгоритмом регіонального розподілення вакцин.

Далі по ефективності розташовується алгоритм критичного періоду. Порівнюючи цей алгоритм та попередній, у випадку, коли алгоритм критичного періоду вибирав місяць вакцинації той самий який і у випадку з пропорційним алгоритмом, то гранична вигода в обох рівна. Це обумовлено тим, що кількість доз вакцин в обох алгоритмах рівна. У всіх інших випадках, другий алгоритм має кращі результати. Лише 2 регіони мають нульову граничну вигоду. Це результат того, що крива цих регіонів почала зростати лише в кінці року, а вакцинація у кінці року не впливає на кількість хворих у регіоні.

Наступним є пропорційний алгоритм з оптимізованим часом. Тут так само кількість доз вакцин така ж як і в перший двох алгоритмах, але за рахунок оптимізації по часу, даний алгоритм виграє над першим (від 50% до 110%) та другим (від 1% до 20%).

Далі будуть розглянуті всі види прогностичного алгоритму. При кроці в 0.1 обидва види прогностичного алгоритму (вакцинація один раз та два рази) програють оптимізованому по часу пропорційному алгоритму, але виграють у першого та другого алгоритмів.

Але при зменшені кроку а значення 0.0001, обидві варіації прогностичного алгоритму виграють у всіх раніше описаних алгоритмах. Всі регіони отримали граничну вигоду більшу ніж в інших алгоритмах.

При порівнянні прогностичного алгоритму з одним або двома вакцинаціями за весь період, варіація алгоритму з одним разом вакцинацією виграє на декілька відсотків.

Отже по результатам розрахунків видно, що прогностичний алгоритм виграє перед усіма іншими алгоритмами.

Таблиця 7.15 – Граничні вигоди пропорційного та оптимізованого пропорційного алгоритмів

|  |  |  |
| --- | --- | --- |
| Найменування регіону | Методи | |
| Пропорційний | Пропорційний з оптимізацією по часу |
| Вінницька | 11.90098998 | 18.90237315 |
| Волинська | 0.112175052 | 0.281644803 |
| Дніпропетровська | 11.6475281 | 16.69523306 |
| Донецька | 12.2295481 | 16.98842535 |
| Житомирська | 94.06229662 | 152.4278078 |
| Закарпатська | 0.103201141 | 0.182161757 |
| Запорізька | 4.073154635 | 6.790216558 |
| Ів.-Франківська | 1.048959367 | 1.968908939 |
| Київська | 21.69776991 | 33.85025318 |
| Кіровоградська | 44.39218353 | 73.76633817 |
| Луганська | 6.275904784 | 9.367146233 |
| Львівська | 36.18099761 | 53.70481436 |
| Миколаївська | 10.89431364 | 19.19758653 |
| Одеська | 25.56458564 | 38.15057158 |
| Полтавська | 4.858043161 | 9.969622048 |
| Рівненська | 81.46204996 | 133.097479 |
| Сумська | 44.5161687 | 77.01942708 |
| Тернопільська | 4.359572671 | 7.748284257 |
| Харківська | 1.289014811 | 2.888894849 |
| Херсонська | 11.42980429 | 18.65779412 |
| Хмельницька | 19.63085036 | 33.96410387 |
| Черкаська | 1.610627637 | 3.62933001 |
| Чернівецька | 6.22967225 | 11.28303988 |
| Чернігівська | 23.58220131 | 42.41469311 |
| м. Київ | 45.87325871 | 67.15924129 |

Таблиця 7.16 – Граничні вигоди критичного періоду та оптимізованого пропорційного алгоритмів.

|  |  |  |
| --- | --- | --- |
| Найменування регіону | Методи | |
| Критичний період | Пропорційний з оптимізацією по часу |
| Вінницька | 11.90098998 | 18.90237315 |
| Волинська | 0.185477273 | 0.281644803 |
| Дніпропетровська | 15.63479454 | 16.69523306 |
| Донецька | 12.2295481 | 16.98842535 |
| Житомирська | 151.7887533 | 152.4278078 |
| Закарпатська | 0.168002863 | 0.182161757 |
| Запорізька | 6.280204658 | 6.790216558 |
| Ів.-Франківська | 1.726196808 | 1.968908939 |
| Київська | 33.81568094 | 33.85025318 |
| Кіровоградська | 58.58957132 | 73.76633817 |
| Луганська | 9.357606297 | 9.367146233 |
| Львівська | 51.92120335 | 53.70481436 |
| Миколаївська | 17.9254741 | 19.19758653 |
| Одеська | 0 | 38.15057158 |
| Полтавська | 8.701601082 | 9.969622048 |
| Рівненська | 107.5120471 | 133.097479 |
| Сумська | 73.96332722 | 77.01942708 |
| Тернопільська | 7.565956465 | 7.748284257 |
| Харківська | 2.368272347 | 2.888894849 |
| Херсонська | 0 | 18.65779412 |
| Хмельницька | 32.47845337 | 33.96410387 |
| Черкаська | 2.640335819 | 3.62933001 |
| Чернівецька | 11.04606399 | 11.28303988 |
| Чернігівська | 39.15339299 | 42.41469311 |
| м. Київ | 64.5506061 | 67.15924129 |

Таблиця 7.17 – Граничні вигоди прогностичного алгоритму з однією вакцинацією з кроком в 0.1 та 0.0001.

|  |  |  |
| --- | --- | --- |
| Найменування регіону | Методи | |
| Прогностичний, крок = 0.1, кількість вакцинацій = 1 | Прогностичний, крок = 0.0001, кількість вакцинацій = 1 |
| Вінницька | 16.5673375 | 22.05819579 |
| Волинська | 0.231082571 | 0.302647038 |
| Дніпропетровська | 16.02346183 | 21.51459985 |
| Донецька | 16.96822176 | 22.71900561 |
| Житомирська | 129.910278 | 173.0985155 |
| Закарпатська | 0.142773026 | 0.205305834 |
| Запорізька | 5.60390385 | 8.006928215 |
| Ів.-Франківська | 1.554149706 | 2.224348893 |
| Київська | 30.00522224 | 40.21426791 |
| Кіровоградська | 61.19765519 | 81.46326698 |
| Луганська | 8.63425668 | 11.54066452 |
| Львівська | 49.89063008 | 67.22785385 |
| Миколаївська | 15.11892222 | 21.44510602 |
| Одеська | 35.16779906 | 47.34187514 |
| Полтавська | 7.989356772 | 11.24355198 |
| Рівненська | 112.9756973 | 149.9520202 |
| Сумська | 64.15825734 | 86.20457245 |
| Тернопільська | 6.302197766 | 8.615846572 |
| Харківська | 2.57555011 | 3.584785882 |
| Херсонська | 15.72347184 | 20.78365006 |
| Хмельницька | 27.57038083 | 38.51326184 |
| Черкаська | 2.896381588 | 4.010567586 |
| Чернівецька | 9.107173573 | 12.36620923 |
| Чернігівська | 32.84809347 | 46.82441194 |
| м. Київ | 63.51377767 | 84.32084766 |

Таблиця 7.18 – Граничні вигоди прогностичного алгоритму з двома вакцинаціями з кроком в 0.1 та 0.0001.

|  |  |  |
| --- | --- | --- |
| Найменування регіону | Методи | |
| Прогностичний, крок = 0.1, кількість вакцинацій = 2 | Прогностичний, крок = 0.0001, кількість вакцинацій = 2 |
| Вінницька | 14.67918701 | 21.88801136 |
| Волинська | 0.213605655 | 0.277428656 |
| Дніпропетровська | 14.23870067 | 21.24193395 |
| Донецька | 15.0547911 | 22.47924367 |
| Житомирська | 115.3381525 | 171.6174379 |
| Закарпатська | 0.128723766 | 0.197240302 |
| Запорізька | 5.013174421 | 7.697478544 |
| Ів.-Франківська | 1.436605612 | 2.093705326 |
| Київська | 26.64119863 | 39.75028975 |
| Кіровоградська | 54.34872739 | 80.81933938 |
| Луганська | 7.672332215 | 11.42128715 |
| Львівська | 44.31321408 | 66.33055717 |
| Миколаївська | 13.39545646 | 20.70918044 |
| Одеська | 31.25117319 | 46.71140418 |
| Полтавська | 7.385000469 | 10.55877954 |
| Рівненська | 100.1478086 | 149.0126324 |
| Сумська | 56.56866137 | 85.48539046 |
| Тернопільська | 5.557340175 | 8.477980058 |
| Харківська | 2.380755321 | 3.297738437 |
| Херсонська | 13.97230158 | 20.67202225 |
| Хмельницька | 24.38724529 | 37.54372288 |
| Черкаська | 2.677200144 | 3.676211253 |
| Чернівецька | 8.030502841 | 12.18871655 |
| Чернігівська | 29.29762149 | 45.01268379 |
| м. Київ | 56.35679066 | 83.76863442 |

Таблиця 7.19 – Граничні вигоди прогностичного алгоритму з кроком в 0.1 та кількістю вакцинацій в один раз та два рази за період.

|  |  |  |
| --- | --- | --- |
| Найменування регіону | Методи | |
| Прогностичний, крок = 0.1, кількість вакцинацій = 1 | Прогностичний, крок = 0.1, кількість вакцинацій = 2 |
| Вінницька | 16.5673375 | 14.67918701 |
| Волинська | 0.231082571 | 0.213605655 |
| Дніпропетровська | 16.02346183 | 14.23870067 |
| Донецька | 16.96822176 | 15.0547911 |
| Житомирська | 129.910278 | 115.3381525 |
| Закарпатська | 0.142773026 | 0.128723766 |
| Запорізька | 5.60390385 | 5.013174421 |
| Ів.-Франківська | 1.554149706 | 1.436605612 |
| Київська | 30.00522224 | 26.64119863 |
| Кіровоградська | 61.19765519 | 54.34872739 |
| Луганська | 8.63425668 | 7.672332215 |
| Львівська | 49.89063008 | 44.31321408 |
| Миколаївська | 15.11892222 | 13.39545646 |
| Одеська | 35.16779906 | 31.25117319 |
| Полтавська | 7.989356772 | 7.385000469 |
| Рівненська | 112.9756973 | 100.1478086 |
| Сумська | 64.15825734 | 56.56866137 |
| Тернопільська | 6.302197766 | 5.557340175 |
| Харківська | 2.57555011 | 2.380755321 |
| Херсонська | 15.72347184 | 13.97230158 |
| Хмельницька | 27.57038083 | 24.38724529 |
| Черкаська | 2.896381588 | 2.677200144 |
| Чернівецька | 9.107173573 | 8.030502841 |
| Чернігівська | 32.84809347 | 29.29762149 |
| м. Київ | 63.51377767 | 56.35679066 |

# ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Головною мовою програмування системи регіонального розподілу доз вакцин було обрану мову програмування Java. Java - об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Java є об'єктно-орієнтованою. Всі дані і дії групуються в класи об'єктів. Виключенням з повної об'єктності (як скажімо в Smalltalk) є примітивні типи (int, float тощо). Це було свідомим рішенням проектувальників мови задля збільшення швидкості. Через це Java не вважається повністю об'єктно-орієнтовною мовою.

У Java всі об'єкти є похідними від головного об'єкта (він називається просто Object), з якого вони успадковують базову поведінку і властивості.

У Java можливе тільки одинарне успадкування, завдяки чому виключається можливість конфліктів між членами класу (методи і змінні), які успадковуються від базових класів.

У намірах проектувальників Java мала замінити C++ — об'єктного наступника мови C. Проектувальники почали з аналізу властивостей C++, які є причиною найбільшого числа помилок, щоби створити просту, безпечну і безвідмовну мову програмування.

В Java існує система винятків або ситуацій, коли програма зустрічається з неочікуваними труднощами, наприклад:

* операції над елементом масиву поза його межами або над порожнім елементом
* читання з недоступного каталогу або неправильної адреси URL
* ввід недопустимих даних користувачем

Одна з особливостей концепції віртуальної машини полягає в тому, що помилки (виключення) не призводять до повного краху системи. Крім того, існують інструменти, які «приєднуються» до середовища періоду виконання і кожен раз, коли сталося певне виключення, записують інформацію з пам'яті для зневадження програми. Ці інструменти автоматизованої обробки виключень надають основну інформацію щодо виключень в програмах на Java.

Java використовує автоматичний збирач сміття для керування пам'яттю під час життєвого циклу об'єкта. Програміст вирішує, коли створювати об'єкти, а віртуальна машина відповідальна за звільнення пам'яті після того, як об'єкт стає непотрібним. Коли до певного об'єкта вже не залишається посилань, збирач сміття може автоматично прибирати його із пам'яті. Проте, витік пам'яті все ж може статися, якщо код, написаний програмістом, має посилання на вже непотрібні об'єкти, наприклад на об'єкти, що зберігаються у діючих контейнерах.

Збирання сміття дозволене у будь-який час. В ідеалі воно відбувається під час бездіяльності програми. Збірка сміття автоматично форсується при нестачі вільної пам'яті в купі для розміщення нового об'єкта, що може призводити до кількасекундного зависання. Тому існують реалізації віртуальної машини Java з прибиральником сміття, спеціально створеним для програмування систем реального часу.

Для знаходження розрахункових був використаний Excel. Microsoft Excel (повна назва Microsoft Office Excel)  —  табличний процесор, програма для роботи з електронними таблицями, створена корпорацією Microsoft для Microsoft Windows, Windows NT і Mac OS. Програма входить до складу офісного пакету Microsoft Office.

В табличному процесорі Excel був використаний додаток Solved Add-in. "Розв’язувач" – це надбудова Microsoft Excel, яка використовується для аналіз "what-if". За її допомогою можна знайти оптимальне (максимальне або мінімальне) значення для формула в одній клітинці (так званій клітинці цільової функції), що обмежується значеннями формул в інших клітинках аркуша. Надбудова "Розв’язувач" працює із групою клітинок (які називаються клітинками змінних рішення або просто клітинками змінних), що використовуються для обчислення формул у цільових функціях і клітинках обмежень. Надбудова регулює значення у клітинках змінних відповідно до меж у клітинках обмежень і виводить потрібний результат у клітинці цільової функції.

Простіше кажучи, ви можете використовувати Розв’язувач, щоб визначати максимальне або мінімальне значення однієї клітинки, змінюючи інші клітинки. Наприклад, можна змінити прогнозований бюджет на рекламу та побачити, яке це вплине на прогнозований прибуток.

В цьому додатку був використаний градієнтний метод для знаходження масиву .

Для створення графічної графічної частини програмного забезпечення був використаний JavaFX. JavaFX — платформа та набір інструментів для створення насичених інтернет-застосунків (англ. Rich Internet Applications, RIA) з можливістю підвантаження медіа та змісту. Вперше продемонстровано Sun Microsystems на Міжнародній конференції Java-розробників JavaOne у травні 2007. JavaFX включає в себе набір утиліт, за допомогою яких веб-розробники та дизайнери можуть швидко створювати та надавати розвинуті інтернет-застосунки для десктопів, мобільних пристроїв, телебачення та інших платформ.

JavaFX складається з JavaFX Script і JavaFX Mobile. Починаючи з випуску JavaFX 2.0 забезпечено можливість створення JavaFX-застосунків, написаних цілком мовою Java. Для розробки застосунків доступний багатий графічний і мультимедійний API, що спрощує створення візуальних програм.

# МАЙБУТНЯ РОБОТА

В процесі розробки алгоритмів оптимального фармацевтичного забезпечення грипозними вакцинами було зроблено важливе припущення, що всі регіони незалежні один від одного. Тобто було припущено, що ефект вакцинацій в одному регіоні не впливає на епідемічний процес грипу в жодному з інших регіонів, що розглядаються. Це надмірне спрощення реальної епідемічної системи. Дійсно, цілком імовірно, що вакцинація в одному регіоні матиме додатковий позитивний вплив на сусідні регіони. Розгляд цих вторинних ефектів має потенціал для підвищення ефективності розроблюваних алгоритмів у майбутньому.

ВИСНОВКИ

Метою даної є вдосконалення математичної моделі, методів та алгоритмів регіонального розподілу вакцин для досягнення балансу між оптимальним розподіленням вакцин та максимальною маргінальною вигодою від одної вакцини.

У процесі виконання роботи було вивчено і проаналізовано предметну область, що стосується теми дипломної роботи, та проаналізовано підходи до регіонального розподілу доз вакцин, розглянутий градієнтний метод для отримання параметрів передачі збудника.

Було проаналізовано та порівняно існуючі системи регіонального розподілу доз вакцин.

Здійснено аналіз математичних методів та обрано метод, який підходить для вирішення поставленої задачі. Визначено дані, по яким буде працювати тестова система. Система регіонального розподілу доз вакцин представляє собою математичне та програмне забезпечення, що надає можливість користувачу отримати оптимальний розподіл доз вакцин по регіонам.

Розроблено та протестовано систему, що надає такі можливості:

* ведення даних користувача;
* побудова моделі оптимального регіонального розподілення доз вакцин на основі таких даних: загальна кількість осіб в кожному регіоні, початкова кількість хворих в кожному регіоні, параметри збудника для кожного регіону.

Було обрано ряд інструментів для реалізації дипломного проекту. Обрано мову програмування Java, бібліотеку JavaFX та програму Excel.

Здійснено випробування, в результаті яких можливо сказати, що програмне забезпечення працює коректно.

У подальшому розвитку даної теми можна запропонувати:

* поліпшити систему за рахунок виведення більш коректної граничної вигоди;
* ввести додаткове обмеження по кількості доступних дох вакцин.

Перелік посилань

1. Hardelid P, Fleming DM, McMenamin J, Andrews N, Robertson C, Sebastian Pillai P. Effectiveness of pandemic and seasonal influenza vaccine in preventing pandemic influenza A(H1N1)2009 infection in England and Scotland 2009–2010. 2010. – 2183 p.
2. Centers for Disease Control and Prevention Selecting the viruses in the seasonal influenza (flu) vaccine. 2012. – 15 p.
3. Hill AN, Longini IM. The critical vaccination fraction for heterogeneous epidemic models. 2012. – 250 p.
4. Finkelstein SN, Hedberg KJ, Hopkins JA, Hashmi S, Larson RC. Vaccine availability in the United States during the 2009 H1N1 outbreak. 2012. – 238 p.
5. Chowell G, Viboud C, Wang X, Bertozzi S, Miller M. Adaptive vaccination strategies to mitigate pandemic influenza. 2009. – 201 p.
6. Державна служба статистики України [Електронний ресурс]. – Режим доступу: www.ukrstat.gov.ua – Назва з екрану

ДОДАТОК А  
Лістинг програми

Лістинг файлу NoVaccModel.java

package sample;  
  
public class NoVaccineModel {  
  
 private static double calculateNextS(double s, double i, double betta, double totalPopulation){  
 return s - betta \* i / totalPopulation \* s + i;  
 }  
  
 private static double calculateNextI(double s, double i, double betta, double totalPopulation){  
 return betta \* i / totalPopulation \* s;  
 }  
  
 public static RegionData calculateVacc(RegionData data){  
 RegionData dataTmp = null;  
 try {  
 dataTmp = (RegionData) data.clone();  
 } catch (CloneNotSupportedException e) {  
 e.printStackTrace();  
 }  
 for (int i = 1; i < 12; i++) {  
 dataTmp.getS()[i] = calculateNextS(  
 dataTmp.getS()[i-1],  
 dataTmp.getI()[i-1],  
 dataTmp.getIntensityInfluenzaTransmission()[i-1],  
 dataTmp.getTotalPopulation()  
 );  
  
 dataTmp.getI()[i] = calculateNextI(  
 dataTmp.getS()[i-1],  
 dataTmp.getI()[i-1],  
 dataTmp.getIntensityInfluenzaTransmission()[i-1],  
 dataTmp.getTotalPopulation()  
 );  
 }  
  
 return dataTmp;  
 }  
}

Лістинг файлу ProportionalVacc.java

package sample;  
  
public class ProportionalVacc {  
  
 private double totalPopulation;  
  
 public ProportionalVacc(double totalPopulation){  
 this.totalPopulation = totalPopulation;  
 }  
  
 public double[] findBestMonth(RegionData data){  
 double bestGV = 0;  
 double[] bestVacc = new double[12];  
  
 for (int i = 0; i < 12; i++) {  
 RegionData dataNoVacc = null;  
 RegionData dataVacc = null;  
 try {  
 dataNoVacc = (RegionData) data.clone();  
 dataVacc = (RegionData) data.clone();  
 } catch (CloneNotSupportedException e) {  
 e.printStackTrace();  
 }  
  
 double[] vaccParam = new double[12];  
 vaccParam[i] = data.getTotalPopulation() / totalPopulation;  
 dataVacc.setVaccParam(vaccParam);  
 RegionData resultNoVacc = NoVaccineModel.calculateVacc(dataNoVacc);  
 RegionData resultVacc = VaccineModel.calculateVacc(dataVacc);  
 double gv = calculateGV(resultNoVacc, resultVacc);  
  
 if (gv > bestGV){  
 bestGV = gv;  
 bestVacc = vaccParam.clone();  
 }  
 }  
  
 return bestVacc;  
 }  
  
 public double calculateGV(RegionData noVacc, RegionData vacc){  
 double diffS = 0;  
 double totalV = 0;  
  
 for (int i = 0; i < noVacc.getS().length; i++) {  
 diffS += Math.abs(noVacc.getI()[i] - vacc.getI()[i]);  
 totalV += vacc.getV()[i];  
 }  
  
 return (diffS / totalV) \* vacc.getTotalPopulation();  
 }  
}

Лістинг файлу CriticalVacc.java

package sample;  
  
public class CriticalVacc {  
  
 private double totalPopulation;  
  
 public CriticalVacc(double totalPopulation) {  
 this.totalPopulation = totalPopulation;  
 }  
  
 public double[] findBestMonth(RegionData data) {  
 double bestGV = 0;  
 double[] bestVacc = new double[12];  
  
 for (int i = 1; i < 12; i++) {  
 RegionData dataNoVacc = null;  
 RegionData dataVacc = null;  
 try {  
 dataNoVacc = (RegionData) data.clone();  
 dataVacc = (RegionData) data.clone();  
 } catch (CloneNotSupportedException e) {  
 e.printStackTrace();  
 }  
 RegionData resultNoVacc = NoVaccineModel.calculateVacc(dataNoVacc);  
 if (resultNoVacc.getI()[i] > resultNoVacc.getI()[i-1]){  
 double[] vaccParam = new double[12];  
 vaccParam[i] = data.getTotalPopulation() / totalPopulation;  
 dataVacc.setVaccParam(vaccParam);  
 RegionData resultVacc = VaccineModel.calculateVacc(dataVacc);  
 double gv = calculateGV(resultNoVacc, resultVacc);  
  
 return vaccParam;  
 }  
 }  
  
 return bestVacc;  
 }  
  
 public double calculateGV(RegionData noVacc, RegionData vacc) {  
 double diffS = 0;  
 double totalV = 0;  
  
 for (int i = 0; i < noVacc.getS().length; i++) {  
 diffS += Math.abs(noVacc.getI()[i] - vacc.getI()[i]);  
 totalV += vacc.getV()[i];  
 }  
  
 return (diffS / totalV) \* vacc.getTotalPopulation();  
 }  
}

Лістинг файлу OneTimeVacc.java

package sample;  
  
public class OneTimeVacc {  
  
 public OneTimeVacc(){  
 }  
  
 public double[] findBestMonth(RegionData data){  
 double bestGV = 0;  
 double[] bestVacc = new double[12];  
  
 for (int i = 0; i < 12; i++) {  
 for (int j = 1; j <= 1.0 / data.getVacStep(); j++) {  
 RegionData dataNoVacc = null;  
 RegionData dataVacc = null;  
 try {  
 dataNoVacc = (RegionData) data.clone();  
 dataVacc = (RegionData) data.clone();  
 } catch (CloneNotSupportedException e) {  
 e.printStackTrace();  
 }  
  
 double[] vaccParam = new double[12];  
 vaccParam[i] = (double) j \* data.getVacStep();  
 dataVacc.setVaccParam(vaccParam);  
 RegionData resultNoVacc = NoVaccineModel.calculateVacc(dataNoVacc);  
 RegionData resultVacc = VaccineModel.calculateVacc(dataVacc);  
 double gv = calculateGV(resultNoVacc, resultVacc);  
  
 if (gv > bestGV){  
 bestGV = gv;  
 bestVacc = vaccParam.clone();  
 }  
  
 }  
 }  
  
 return bestVacc;  
 }  
  
 public double calculateGV(RegionData noVacc, RegionData vacc){  
 double diffS = 0;  
 double totalV = 0;  
  
 for (int i = 0; i < noVacc.getS().length; i++) {  
 diffS += Math.abs(noVacc.getI()[i] - vacc.getI()[i]);  
 totalV += vacc.getV()[i];  
 }  
  
 return (diffS / totalV) \* vacc.getTotalPopulation();  
 }  
}

Лістинг файлу TwoTimesVacc.java

package sample;  
  
import java.util.Arrays;  
  
public class TwoTimesVacc {  
  
 public TwoTimesVacc() {  
 }  
  
 public double[] findBestMonth(RegionData data){  
 double bestGV = 0;  
 double[] bestVacc = new double[12];  
  
 double[] arr = new double[12];  
 double max = 1;  
  
 for (double val = data.getVacStep(); val <= max ; val+=data.getVacStep()) {  
 for (int i = 0; i < arr.length; i++) {  
 double[] ar1 = arr.clone();  
// ar1[i] = round(val, 2);  
 ar1[i] = val;  
 for (int j = i+1; j < arr.length-1; j++) {  
 double[] cl2 = ar1.clone();  
// cl2[j] = round(val, 2);  
 cl2[j] = val;  
 RegionData dataNoVacc = null;  
 RegionData dataVacc = null;  
 try {  
 dataNoVacc = (RegionData) data.clone();  
 dataVacc = (RegionData) data.clone();  
 } catch (CloneNotSupportedException e) {  
 e.printStackTrace();  
 }  
  
 dataVacc.setVaccParam(cl2);  
 RegionData resultNoVacc = NoVaccineModel.calculateVacc(dataNoVacc);  
 RegionData resultVacc = VaccineModel.calculateVacc(dataVacc);  
 double gv = calculateGV(resultNoVacc, resultVacc);  
  
 if (gv > bestGV){  
 bestGV = gv;  
 bestVacc = cl2.clone();  
 }  
 }  
 }  
 }  
  
 return bestVacc;  
 }  
  
 public double calculateGV(RegionData noVacc, RegionData vacc){  
 double diffS = 0;  
 double totalV = 0;  
  
 for (int i = 0; i < noVacc.getS().length; i++) {  
 diffS += Math.abs(noVacc.getI()[i] - vacc.getI()[i]);  
 totalV += vacc.getV()[i];  
 }  
  
 return (diffS / totalV) \* vacc.getTotalPopulation();  
 }  
  
 private double round(double value, int places) {  
 if (places < 0) throw new IllegalArgumentException();  
  
 long factor = (long) Math.pow(10, places);  
 value = value \* factor;  
 long tmp = Math.round(value);  
 return (double) tmp / factor;  
 }  
}

Лістинг файлу VaccineModel.java

package sample;  
  
public class VaccineModel {  
  
 private static double calculateNextS(double s, double i, double betta, double totalPopulation, double vParam){  
 return s - betta \* i / totalPopulation\*s + i - vParam \* s;  
 }  
  
 private static double calculateNextI(double s, double i, double betta, double totalPopulation){  
 return betta \* i / totalPopulation \* s;  
 }  
  
 private static double calculateNextV(double s, double v, double vParam){  
 return v + vParam\*s;  
 }  
  
 public static RegionData calculateVacc(RegionData data){  
 RegionData data1 = null;  
  
 try {  
 data1 = (RegionData) data.clone();  
 } catch (CloneNotSupportedException e) {  
 e.printStackTrace();  
 }  
  
 for (int i = 1; i < 12; i++) {  
 data1.getS()[i] = calculateNextS(  
 data1.getS()[i-1],  
 data1.getI()[i-1],  
 data1.getIntensityInfluenzaTransmission()[i-1],  
 data1.getTotalPopulation(),  
 data1.getVaccParam()[i-1]  
 );  
  
 data1.getI()[i] = calculateNextI(  
 data1.getS()[i-1],  
 data1.getI()[i-1],  
 data1.getIntensityInfluenzaTransmission()[i-1],  
 data1.getTotalPopulation()  
 );  
  
 data1.getV()[i] = calculateNextV(  
 data1.getS()[i-1],  
 data1.getV()[i-1],  
 data1.getVaccParam()[i-1]  
 );  
 }  
  
 return data1;  
 }  
}

Лістинг файлу RegionData.java

package sample;  
  
public class RegionData {  
 private String name;  
 private double[] intensityInfluenzaTransmission;  
 private double[] vaccParam;  
 private double totalPopulation;  
 private double startIllNumber;  
 private double startHealthNumber;  
 private double[] s;  
 private double[] i;  
 private double[] v;  
 private double vacStep;  
  
 public static final String[] MONTHS = {  
 "січень", "лютий", "березень",  
 "квітень", "травень", "червень",  
 "липень", "серпень", "вересень",  
 "жовтень", "листопад", "грудень"  
 };  
  
 public RegionData(String name, double[] intensityInfluenzaTransmission, double totalPopulation, double startIllNumber, double vacStep) {  
 this.name = name;  
 this.intensityInfluenzaTransmission = intensityInfluenzaTransmission;  
 this.vacStep = vacStep;  
 this.vaccParam = new double[]{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  
 this.totalPopulation = totalPopulation;  
 this.startIllNumber = startIllNumber;  
 this.startHealthNumber = totalPopulation - startIllNumber;  
 this.s = new double[]{startHealthNumber, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  
 this.i = new double[]{startIllNumber, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  
 this.v = new double[]{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  
 }  
  
 public RegionData(String name, double[] intensityInfluenzaTransmission, double totalPopulation, double startIllNumber, double[] vaccParam, double vacStep) {  
 this.name = name;  
 this.intensityInfluenzaTransmission = intensityInfluenzaTransmission;  
 this.vaccParam = vaccParam;  
 this.totalPopulation = totalPopulation;  
 this.startIllNumber = startIllNumber;  
 this.vacStep = vacStep;  
 this.startHealthNumber = totalPopulation - startIllNumber;  
 this.s = new double[]{startHealthNumber, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  
 this.i = new double[]{startIllNumber, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  
 this.v = new double[]{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public RegionData setName(String name) {  
 this.name = name;  
 return this;  
 }  
  
 public double getVacStep() {  
 return vacStep;  
 }  
  
 public RegionData setVacStep(double vacStep) {  
 this.vacStep = vacStep;  
 return this;  
 }  
  
 public double[] getIntensityInfluenzaTransmission() {  
 return intensityInfluenzaTransmission;  
 }  
  
 public double[] getVaccParam() {  
 return vaccParam;  
 }  
  
 public double getTotalPopulation() {  
 return totalPopulation;  
 }  
  
 public double getStartIllNumber() {  
 return startIllNumber;  
 }  
  
 public double getStartHealthNumber() {  
 return startHealthNumber;  
 }  
  
 public double[] getS() {  
 return s;  
 }  
  
 public double[] getI() {  
 return i;  
 }  
  
 public double[] getV() {  
 return v;  
 }  
  
 public RegionData setIntensityInfluenzaTransmission(double[] intensityInfluenzaTransmission) {  
 this.intensityInfluenzaTransmission = intensityInfluenzaTransmission.clone();  
 return this;  
 }  
  
 public RegionData setVaccParam(double[] vaccParam) {  
 this.vaccParam = vaccParam.clone();  
 return this;  
 }  
  
 public RegionData setTotalPopulation(double totalPopulation) {  
 this.totalPopulation = totalPopulation;  
 return this;  
 }  
  
 public RegionData setStartIllNumber(double startIllNumber) {  
 this.startIllNumber = startIllNumber;  
 return this;  
 }  
  
 public RegionData setStartHealthNumber(double startHealthNumber) {  
 this.startHealthNumber = startHealthNumber;  
 return this;  
 }  
  
 public RegionData setS(double[] vacS) {  
 this.s = vacS;  
 return this;  
 }  
  
 public RegionData setI(double[] vacI) {  
 this.i = vacI;  
 return this;  
 }  
  
 public RegionData setV(double[] vacV) {  
 this.v = vacV;  
 return this;  
 }  
  
 @Override  
 protected Object clone() throws CloneNotSupportedException {  
 return new RegionData(  
 name,  
 this.intensityInfluenzaTransmission.clone(),  
 this.totalPopulation,  
 this.startIllNumber,  
 this.vaccParam.clone(),  
 vacStep);  
 }  
  
 @Override  
 public String toString() {  
 StringBuilder result = new StringBuilder();  
  
 for (int i = 0; i < 12; i++) {  
 result.append(String.format("%s: \ts = %.15f, \ti = %.15f\n", MONTHS[i], s[i], this.i[i]));  
 }  
  
 return result.toString();  
 }  
}

Лістинг файлу Main.java

package sample;  
  
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.stage.Stage;  
  
import java.util.\*;  
  
public class Main extends Application {  
  
 public static List<RegionData> regions = new ArrayList<>();  
  
 public static void main(String[] args) {  
 launch(args);  
 }  
  
 @Override  
 public void start(Stage primaryStage) throws Exception{  
 Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));  
 primaryStage.setTitle("Hello World");  
 primaryStage.setResizable(false);  
 primaryStage.setScene(new Scene(root, 1200, 800));  
 primaryStage.show();  
 }  
  
}

Лістинг файлу Controller.java

package sample;  
  
import javafx.event.ActionEvent;  
import javafx.event.Event;  
import javafx.fxml.Initializable;  
import javafx.geometry.Pos;  
import javafx.scene.Node;  
import javafx.scene.chart.\*;  
import javafx.scene.control.\*;  
import javafx.scene.layout.AnchorPane;  
import javafx.scene.layout.VBox;  
import javafx.scene.text.Font;  
  
import java.math.BigDecimal;  
import java.math.RoundingMode;  
import java.net.URL;  
import java.util.Arrays;  
import java.util.ResourceBundle;  
  
public class Controller implements Initializable {  
 public AnchorPane bettaAnchor;  
 public AnchorPane vaccAnchor;  
 public TextField regionName;  
 public TextField population;  
 public TextField illNumber;  
 public TextField vaccStep;  
 public Button partVaccCalc;  
 public Button criticalVaccCalc;  
 public Button oneTimeVaccCalc;  
 public Button twoTimeVaccCalc;  
 public VBox partitionVBox;  
 public VBox criticalVBox;  
 public VBox oneTimeVBox;  
 public VBox twoTimeVBox;  
  
 double step;  
  
 public void addRegion() {  
 double[] betta = new double[12];  
 double[] vParam = new double[12];  
 String regionNameValue;  
 long populationValue;  
 double illNumberValue;  
 double vacStep;  
  
 int counter = 0;  
 for (Node child : bettaAnchor.getChildren()) {  
 if (child.getTypeSelector().toLowerCase().contains("text")) {  
 betta[counter++] = Double.valueOf(((TextField) child).getText());  
 }  
 }  
 counter = 0;  
 for (Node child : vaccAnchor.getChildren()) {  
 if (child.getTypeSelector().toLowerCase().contains("text")) {  
 vParam[counter++] = Double.valueOf(((TextField) child).getText());  
 }  
 }  
  
 regionNameValue = regionName.getText();  
 populationValue = Long.valueOf(population.getText());  
 illNumberValue = Double.valueOf(illNumber.getText());  
 vacStep = Double.valueOf(vaccStep.getText());  
  
 Main.regions.add(new RegionData(regionNameValue, betta, populationValue, illNumberValue, vParam, vacStep));  
 }  
  
 public void clearRegions(ActionEvent event) {  
 }  
  
 public void calculate1TimeVacc(ActionEvent event) {  
 oneTimeVBox.getChildren().clear();  
  
 int counter = 0;  
  
 for (RegionData region : Main.regions) {  
 OneTimeVacc oneTimeVacc = new OneTimeVacc();  
 double[] bestMonth = oneTimeVacc.findBestMonth(region);  
 RegionData noVacData = NoVaccineModel.calculateVacc(region);  
  
 RegionData dataTmp = null;  
 try {  
 dataTmp = (RegionData) region.clone();  
 } catch (CloneNotSupportedException e) {  
 e.printStackTrace();  
 }  
  
 dataTmp.setVaccParam(bestMonth);  
  
 RegionData vacData = VaccineModel.calculateVacc(dataTmp);  
  
 final CategoryAxis xAxis = new CategoryAxis();  
 final NumberAxis yAxis = new NumberAxis();  
 xAxis.setLabel("Month");  
 yAxis.setLabel("Number of people");  
 LineChart<String, Number> chart = new LineChart<>(xAxis, yAxis);  
 XYChart.Series<String, Number> seriesNoVac = new XYChart.Series<>();  
 seriesNoVac.setName("No vacc");  
 XYChart.Series<String, Number> seriesVac = new XYChart.Series<>();  
 seriesVac.setName("Vacc");  
  
 for (int i = 0; i < 12; i++) {  
 seriesNoVac.getData().add(new XYChart.Data<>(RegionData.MONTHS[i], noVacData.getI()[i]));  
 seriesVac.getData().add(new XYChart.Data<>(RegionData.MONTHS[i], vacData.getI()[i]));  
 }  
  
 chart.getData().addAll(seriesNoVac, seriesVac);  
 chart.setLayoutX(5.0);  
 chart.setLayoutY(550 \* counter++);  
 chart.setMinHeight(500.0);  
 chart.setMinWidth(1000);  
  
 Label label = new Label(region.getName());  
 label.setMinWidth(250);  
 label.setFont(Font.font(30));  
 label.setAlignment(Pos.CENTER);  
  
 Label mbLabel = new Label("Marginal benefit = " + oneTimeVacc.calculateGV(noVacData, vacData) \* 100\_000);  
 mbLabel.setMinWidth(250);  
 mbLabel.setFont(Font.font(20));  
 mbLabel.setAlignment(Pos.CENTER);  
  
 StringBuilder str = new StringBuilder("[");  
 for (int i = 0; i < 12; i++) {  
 str.append(new BigDecimal(bestMonth[i]).setScale(5, RoundingMode.CEILING).toString());  
 if (i != 11) {  
 str.append(", ");  
 }  
 }  
 str.append("]");  
  
 Label vaccArr = new Label("Vacc model = " + str);  
 vaccArr.setMinWidth(500);  
 vaccArr.setFont(Font.font(20));  
 vaccArr.setAlignment(Pos.CENTER);  
  
 oneTimeVBox.getChildren().add(label);  
 oneTimeVBox.getChildren().add(mbLabel);  
 oneTimeVBox.getChildren().add(vaccArr);  
 oneTimeVBox.getChildren().add(chart);  
  
 System.out.println(Arrays.toString(bestMonth));  
 }  
 }  
  
 @Override  
 public void initialize(URL location, ResourceBundle resources) {  
 setTextFormator();  
 }  
  
 private void setTextFormator() {  
 for (Node child : bettaAnchor.getChildren()) {  
 if (child.getTypeSelector().toLowerCase().contains("text")) {  
 ((TextField) child).textProperty().addListener((observable, oldValue, newValue) -> {  
 if (!newValue.matches("\\d{0,10}([\\.]\\d{0,15})?")) {  
 ((TextField) child).setText(oldValue);  
 }  
 });  
 }  
 }  
 for (Node child : vaccAnchor.getChildren()) {  
 if (child.getTypeSelector().toLowerCase().contains("text")) {  
 ((TextField) child).textProperty().addListener((observable, oldValue, newValue) -> {  
 if (!newValue.matches("0+(\\.\\d+)?|1\\.0") && !newValue.matches("^0+\\.$") && !newValue.matches("^1$") && !newValue.matches("^1\\.$") && !newValue.matches("^\\s\*$")) {  
 ((TextField) child).setText(oldValue);  
 }  
 });  
 }  
  
 }  
 population.textProperty().addListener((observable, oldValue, newValue) -> {  
 if (!newValue.matches("\\d{0,15}?")) {  
 population.setText(oldValue);  
 }  
 });  
 illNumber.textProperty().addListener((observable, oldValue, newValue) -> {  
 if (!newValue.matches("\\d{0,10}([\\.]\\d{0,15})?")) {  
 illNumber.setText(oldValue);  
 }  
 });  
 vaccStep.textProperty().addListener((observable, oldValue, newValue) -> {  
 if (!newValue.matches("\\d{0,10}([\\.]\\d{0,15})?")) {  
 vaccStep.setText(oldValue);  
 }  
 });  
 }  
  
 public void disableBottomBar(Event event) {  
 oneTimeVaccCalc.setDisable(true);  
 twoTimeVaccCalc.setDisable(true);  
 partVaccCalc.setDisable(true);  
 }  
  
 public void enableBottomBar(Event event) {  
 if (oneTimeVaccCalc != null) {  
 oneTimeVaccCalc.setDisable(false);  
 }  
 if (twoTimeVaccCalc != null) {  
 twoTimeVaccCalc.setDisable(false);  
 }  
 if (partVaccCalc != null) {  
 partVaccCalc.setDisable(false);  
 }  
 }  
  
 public void calculate2TimeVacc(ActionEvent event) {  
 twoTimeVBox.getChildren().clear();  
  
 int counter = 0;  
  
 for (RegionData region : Main.regions) {  
 TwoTimesVacc twoTimesVacc = new TwoTimesVacc();  
 double[] bestMonth = twoTimesVacc.findBestMonth(region);  
 RegionData noVacData = NoVaccineModel.calculateVacc(region);  
  
 RegionData dataTmp = null;  
 try {  
 dataTmp = (RegionData) region.clone();  
 } catch (CloneNotSupportedException e) {  
 e.printStackTrace();  
 }  
  
 dataTmp.setVaccParam(bestMonth);  
  
 RegionData vacData = VaccineModel.calculateVacc(dataTmp);  
  
 final CategoryAxis xAxis = new CategoryAxis();  
 final NumberAxis yAxis = new NumberAxis();  
 xAxis.setLabel("Month");  
 yAxis.setLabel("Number of people");  
 LineChart<String, Number> chart = new LineChart<>(xAxis, yAxis);  
 XYChart.Series<String, Number> seriesNoVac = new XYChart.Series<>();  
 seriesNoVac.setName("No vacc");  
 XYChart.Series<String, Number> seriesVac = new XYChart.Series<>();  
 seriesVac.setName("Vacc");  
  
 for (int i = 0; i < 12; i++) {  
 seriesNoVac.getData().add(new XYChart.Data<>(RegionData.MONTHS[i], noVacData.getI()[i]));  
 seriesVac.getData().add(new XYChart.Data<>(RegionData.MONTHS[i], vacData.getI()[i]));  
 }  
  
 chart.getData().addAll(seriesNoVac, seriesVac);  
 chart.setLayoutX(5.0);  
 chart.setLayoutY(550 \* counter++);  
 chart.setMinHeight(500.0);  
 chart.setMinWidth(1000);  
  
 Label label = new Label(region.getName());  
 label.setMinWidth(250);  
 label.setFont(Font.font(30));  
 label.setAlignment(Pos.CENTER);  
  
 Label mbLabel = new Label("Marginal benefit = " + twoTimesVacc.calculateGV(noVacData, vacData) \* 100\_000);  
 mbLabel.setMinWidth(250);  
 mbLabel.setFont(Font.font(20));  
 mbLabel.setAlignment(Pos.CENTER);  
  
 Label vaccArr = new Label("Vacc model = " + Arrays.toString(bestMonth));  
 vaccArr.setMinWidth(250);  
 vaccArr.setFont(Font.font(20));  
 vaccArr.setAlignment(Pos.CENTER);  
  
 twoTimeVBox.getChildren().add(label);  
 twoTimeVBox.getChildren().add(mbLabel);  
 twoTimeVBox.getChildren().add(vaccArr);  
 twoTimeVBox.getChildren().add(chart);  
  
// System.out.println(region.getName() + ": " + Arrays.toString(bestMonth));  
 System.out.println(twoTimesVacc.calculateGV(noVacData, vacData) \* 100\_000);  
 }  
 }  
  
 public void calculatePartVacc(ActionEvent event) {  
 twoTimeVBox.getChildren().clear();  
  
 int counter = 0;  
  
  
 for (RegionData region : Main.regions) {  
 ProportionalVacc proportionalVacc = new ProportionalVacc(Main.regions.stream().mapToDouble(RegionData::getTotalPopulation).sum());  
 double[] bestMonth = proportionalVacc.findBestMonth(region);  
 RegionData noVacData = NoVaccineModel.calculateVacc(region);  
  
 RegionData dataTmp = null;  
 try {  
 dataTmp = (RegionData) region.clone();  
 } catch (CloneNotSupportedException e) {  
 e.printStackTrace();  
 }  
  
 dataTmp.setVaccParam(bestMonth);  
  
 RegionData vacData = VaccineModel.calculateVacc(dataTmp);  
  
 final CategoryAxis xAxis = new CategoryAxis();  
 final NumberAxis yAxis = new NumberAxis();  
 xAxis.setLabel("Month");  
 yAxis.setLabel("Number of people");  
 LineChart<String, Number> chart = new LineChart<>(xAxis, yAxis);  
 XYChart.Series<String, Number> seriesNoVac = new XYChart.Series<>();  
 seriesNoVac.setName("No vacc");  
 XYChart.Series<String, Number> seriesVac = new XYChart.Series<>();  
 seriesVac.setName("Vacc");  
  
 for (int i = 0; i < 12; i++) {  
 seriesNoVac.getData().add(new XYChart.Data<>(RegionData.MONTHS[i], noVacData.getI()[i]));  
 seriesVac.getData().add(new XYChart.Data<>(RegionData.MONTHS[i], vacData.getI()[i]));  
 }  
  
 chart.getData().addAll(seriesNoVac, seriesVac);  
 chart.setLayoutX(5.0);  
 chart.setLayoutY(550 \* counter++);  
 chart.setMinHeight(500.0);  
 chart.setMinWidth(1000);  
  
 Label label = new Label(region.getName());  
 label.setMinWidth(250);  
 label.setFont(Font.font(30));  
 label.setAlignment(Pos.CENTER);  
  
 Label mbLabel = new Label("Marginal benefit = " + proportionalVacc.calculateGV(noVacData, vacData) \* 100\_000);  
 mbLabel.setMinWidth(250);  
 mbLabel.setFont(Font.font(20));  
 mbLabel.setAlignment(Pos.CENTER);  
  
 Label vaccArr = new Label("Vacc model = " + Arrays.toString(bestMonth));  
 vaccArr.setMinWidth(250);  
 vaccArr.setFont(Font.font(20));  
 vaccArr.setAlignment(Pos.CENTER);  
  
 partitionVBox.getChildren().add(label);  
 partitionVBox.getChildren().add(mbLabel);  
 partitionVBox.getChildren().add(vaccArr);  
 partitionVBox.getChildren().add(chart);  
 }  
 }  
  
 public void calculateCriticalVacc(ActionEvent actionEvent) {  
 twoTimeVBox.getChildren().clear();  
  
 int counter = 0;  
  
  
 for (RegionData region : Main.regions) {  
 CriticalVacc criticalVacc = new CriticalVacc(Main.regions.stream().mapToDouble(RegionData::getTotalPopulation).sum());  
 double[] bestMonth = criticalVacc.findBestMonth(region);  
 RegionData noVacData = NoVaccineModel.calculateVacc(region);  
  
 RegionData dataTmp = null;  
 try {  
 dataTmp = (RegionData) region.clone();  
 } catch (CloneNotSupportedException e) {  
 e.printStackTrace();  
 }  
  
 dataTmp.setVaccParam(bestMonth);  
  
 RegionData vacData = VaccineModel.calculateVacc(dataTmp);  
  
 final CategoryAxis xAxis = new CategoryAxis();  
 final NumberAxis yAxis = new NumberAxis();  
 xAxis.setLabel("Month");  
 yAxis.setLabel("Number of people");  
 LineChart<String, Number> chart = new LineChart<>(xAxis, yAxis);  
 XYChart.Series<String, Number> seriesNoVac = new XYChart.Series<>();  
 seriesNoVac.setName("No vacc");  
 XYChart.Series<String, Number> seriesVac = new XYChart.Series<>();  
 seriesVac.setName("Vacc");  
  
 for (int i = 0; i < 12; i++) {  
 seriesNoVac.getData().add(new XYChart.Data<>(RegionData.MONTHS[i], noVacData.getI()[i]));  
 seriesVac.getData().add(new XYChart.Data<>(RegionData.MONTHS[i], vacData.getI()[i]));  
 }  
  
 chart.getData().addAll(seriesNoVac, seriesVac);  
 chart.setLayoutX(5.0);  
 chart.setLayoutY(550 \* counter++);  
 chart.setMinHeight(500.0);  
 chart.setMinWidth(1000);  
  
 Label label = new Label(region.getName());  
 label.setMinWidth(250);  
 label.setFont(Font.font(30));  
 label.setAlignment(Pos.CENTER);  
  
 Label mbLabel = new Label("Marginal benefit = " + criticalVacc.calculateGV(noVacData, vacData) \* 100\_000);  
 mbLabel.setMinWidth(250);  
 mbLabel.setFont(Font.font(20));  
 mbLabel.setAlignment(Pos.CENTER);  
  
 Label vaccArr = new Label("Vacc model = " + Arrays.toString(bestMonth));  
 vaccArr.setMinWidth(250);  
 vaccArr.setFont(Font.font(20));  
 vaccArr.setAlignment(Pos.CENTER);  
  
 criticalVBox.getChildren().add(label);  
 criticalVBox.getChildren().add(mbLabel);  
 criticalVBox.getChildren().add(vaccArr);  
 criticalVBox.getChildren().add(chart);  
 }  
 }  
}

ДОДАТОК Б  
Ілюстрований матеріал