

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ІНФОРМАТИКИ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Курсова робота з дисципліни «Моделювання систем»

Тема: Імітаційна модель роботи регулювальної ділянки цеху

на основі формалізму мережа Петрі

|  |  |
| --- | --- |
| **Керівник**:  асистент кафедри ІПІ  Дифучина Олександра Юріївна  «Допущено до захисту»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (особистий підпис керівника)  «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023 р.  Захищено з оцінкою  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Члени комісії:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | **Виконавець:**  Чапча Святослав Олександрович  студент ІV курсу  групи ІТ-04  залікова книжка № ІТ-0425  «25» грудня 2023 р.  Інна СТЕЦЕНКО  Олександра ДИФУЧИНА |

Київ 2023

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра інформатики та програмної інженерії

Дисципліна «Моделювання систем»

Спеціальність Інженерія програмного забезпечення

Курс 4 Група ІТ – 04 Семестр 7

ЗАВДАННЯ

на курсову роботу студента

Чапчи Святослава Олександровича

(прізвище, ім'я, по батькові)

1.Тема роботи: Імітаційна модель роботи регулювальної ділянки цеху

на основі формалізму мережа Петрі

2. Термін здачі студентом закінченої роботи "25" грудня 2023 р.

3. Вихідні дані до проекту

Завдання № 10 з Навчального Посібника

4. Зміст розрахунково-пояснювальної записки (перелік питань, що розробляються)

Вступ. 1. Розробка концептуальної моделі 2. Розробка формалізованої моделі 3. Програмна реалізація моделі 4. Проведення експериментів 5. Інтерпретація результатів експериментів. Висновки. Список використаних джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Графічного матеріалу не має.

6. Дата видачі завдання "12" вересня 2023 р.

РЕФЕРАТ

Курсова робота: 51с., 12 рис., 5 табл., 2 додатка, 7 джерело літератури.

Об'єкт дослідження – регулювальна ділянка цеху.

Мета роботи – визначення оптимального режиму роботи регулювальної ділянки цеху.

Метод дослідження – імітаційне моделювання роботи регулювальної ділянки. Проведено дослідження різних режимів роботи конвеєра і розроблена програмна реалізація імітаційної моделі системи. Розроблено план і проведені експерименти з імітаційною моделлю. Результати моделювання використані для визначення оптимального режиму роботи регулювальної ділянки.

Ключові слова: ІМІТАЦІЙНА МОДЕЛЬ, МОВА МОДЕЛЮВАННЯ, ПЛАНУВАННЯ ЕКСПЕРИМЕНТІВ, ДИСПЕРСІЙНИЙ АНАЛІЗ, РЕГУЛЮВАЛЬНА ДІЛЯНКА, МЕРЕЖА ПЕТРІ

**ЗМІСТ**

[**ВСТУП** 6](#_Toc153389510)

[**ПОСТАНОВКА ЗАВДАННЯ** 8](#_Toc153389511)

[**РОЗДІЛ 1. КОНЦЕПТУАЛЬНА МОДЕЛЬ** 9](#_Toc153389512)

[1.1 Дослідження можливих видів моделювання 9](#_Toc153389513)

[1.2 Ціль моделювання 10](#_Toc153389514)

[1.3 Концептуальна модель 11](#_Toc153389515)

[1.4 Вхідні та вихідні дані, параметри моделі 11](#_Toc153389516)

[1.5 Обмеження 12](#_Toc153389517)

[1.6 Цільова функція 13](#_Toc153389518)

[**РОЗДІЛ 2. ФОРМАЛІЗОВАНА МОДЕЛЬ** 14](#_Toc153389519)

[2.1 Елементи мережі Петрі 14](#_Toc153389520)

[2.2 Побудова формалізованої моделі 14](#_Toc153389521)

[2.3 Обчислення вихідних характеристики 15](#_Toc153389522)

[**РОЗДІЛ 3. РЕАЛІЗАЦІЯ МОДЕЛІ** 17](#_Toc153389523)

[3.1 Опис програмної реалізації імітаційної моделі 18](#_Toc153389524)

[3.2 Оцінка адекватності моделі 19](#_Toc153389525)

[3.3 Верифікація моделі 21](#_Toc153389526)

[**РОЗДІЛ 4. ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ НА МОДЕЛІ** 23](#_Toc153389527)

[4.1 План експериментів 23](#_Toc153389528)

[4.2 Аналіз і оцінка результатів 23](#_Toc153389529)

[**РОЗДІЛ 5. ІНТЕРПРЕТАЦІЯ РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ** 27](#_Toc153389530)

[**ВИСНОВОК** 28](#_Toc153389531)

[**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** 29](#_Toc153389532)

[**Додаток А. Лістинг коду** 30](#_Toc153389533)

[**Додаток Б. Результати проведення експериментів** 45](#_Toc153389534)

# ВСТУП

В Україні цехи є важливим елементом економіки. Вони забезпечують робочими місцями тисячі людей та виробляють значну частину промислової продукції. У зв'язку з війною та іншими геополітичними факторами зростає потреба в локальному виробництві товарів та послуг. Цехи можуть допомогти забезпечити цю потребу.

Метою дослідження є визначення оптимального режиму роботи регулювальної ділянки цеху.

Для того, щоб вирішити поставлену задачу, потрібно створити модель роботи регулювальної ділянки. Для цього ми використаємо метод імітаційного моделювання.

Модель - це абстрактне відображення об'єкта, системи або концепції, призначене для наукового вивчення. У сучасному світі практично неможливо знайти галузь людського життя, де не використовуються різноманітні методи моделювання.

В загальному випадку модель складається з X - множини вхідних змінних системи, Y - множини вихідних змінних системи, P - множини параметрів та F - функції, функціоналу, алгоритму або формального представлення залежності змінних Y від змінних X.

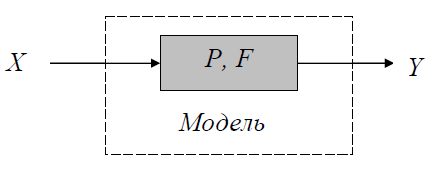


Рисунок 1.1 - Загальна структура моделі

Різні методи моделювання використовуються у всіх сферах людської діяльності і є невід’ємною частиною нашого життя. Основна мета моделювання полягає в тому, щоб знайти значення вихідних змінних Y при відомих значеннях вхідних змінних X, враховуючи відому модель F та визначені параметри P.

Імітаційне моделювання – це метод, який дозволяє досліднику отримати інформацію про властивості реальної системи, багаторазово запускаючи її модель. Імітаційне моделювання враховує зміну властивостей об'єктів у часі, тому їм можна вирішити будь-яке завдання.

Основна перевага імітаційного моделювання – це можливість розробки моделей з мінімальною витратою часу на програмування. Також імітаційне моделювання дозволяє вирішувати складніші завдання, ніж аналітичне дослідження, оскільки воно враховує випадкові дії та інші фактори. Алгоритми імітації мереж Петрі та мереж масового обслуговування, які побудовані на основі універсальних мов програмування, мають високу гнучкість.

Імітаційне моделювання складається з таких етапів:

1. формулювання проблеми та змістовної постановки задачі;
2. розробка концептуальної моделі;
3. розробка імітаційної моделі;
4. оцінка адекватності моделі;
5. планування та проведення експерименту;
6. оцінка точності результатів моделювання;
7. інтерпретація результатів моделювання і прийняття рішення.

Для розв'язання поставленої задачі ми побудуємо концептуальну модель, формалізовану модель із формалізмом мережа Петрі, які відображатимуть роботу регулювальної ділянки. Також ми виконаємо програмну реалізацію моделі зі вказаним формалізмом та проведемо перевірку й експерименти.

# ПОСТАНОВКА ЗАВДАННЯ

На регулювальну ділянку цеху через випадкові інтервали часу надходять по два агрегати в середньому через кожні 30 хвилин. Первинне регулювання здійснюється для двох агрегатів одночасно і займає біля 30 хвилин. Якщо в момент приходу агрегатів попередня партія не була оброблена, агрегати на регулювання не приймаються. Агрегати, які одержали відмову, після первинного регулювання надходять у проміжний накопичувач. З накопичувача агрегати, що пройшли первинне регулювання, надходять попарно на вторинне регулювання, яке виконується в середньому за 30 хвилин, а ті, що не пройшли первинне регулювання, надходять на повне регулювання, що займає 100 хвилин для одного агрегату. Всі величини задані середніми значеннями, розподілені за експоненціальним законом.

Визначити ймовірність відмови в первинному регулюванні і завантаження накопичувача агрегатами, що потребують повного регулювання. Визначити параметри і ввести в систему накопичувач, що забезпечує безвідмовне обслуговування агрегатів, що надходять.

# РОЗДІЛ 1. КОНЦЕПТУАЛЬНА МОДЕЛЬ

## 1.1 Дослідження можливих видів моделювання

Щоб вирішити поставлене завдання, потрібно створити модель системи. Існує багато різних методів моделювання, але найпопулярнішими є аналітичне, імітаційне та статистичне моделювання [1].

Аналітичне моделювання є найстарішим і найпростішим методом моделювання. Воно використовується, коли залежність між вхідними та вихідними змінними системи можна описати аналітичними функціями. Це означає, що можна знайти формули або рівняння, які визначають, як зміняться вихідні змінні системи в залежності від змін вхідних змінних.

Унікальність аналітичного моделювання полягає в тому, що воно дозволяє отримати точніші результати, ніж інші методи моделювання. Це пов'язано з тим, що аналітичні функції зазвичай є точними апроксимаціями реальних систем. Однак аналітичне моделювання має і свої обмеження. Воно може бути застосоване лише до систем, для яких залежність між вхідними та вихідними змінними можна описати аналітичними функціями. Якщо така залежність не існує, то аналітичне моделювання неможливо використовувати.

Імітаційне моделювання є найскладнішим методом моделювання. Воно використовується, коли систему моделюють шляхом її імітації в часі. Цей метод дозволяє враховувати випадковість та інші фактори, що неможливо зробити за допомогою аналітичних або математичних методів [2].

Унікальність імітаційного моделювання полягає в тому, що воно дозволяє моделювати системи, які є занадто складними або непередбачуваними для інших методів моделювання. Це пов'язано з тим, що імітаційне моделювання дозволяє моделювати систему в її реальному середовищі, з урахуванням всіх випадкових факторів, але воно може бути застосоване лише до систем, для яких можна розробити модель імітації. Якщо така модель не може бути розроблена, то імітаційне моделювання неможливо використовувати.

Статистичне моделювання - це метод, який використовується для створення математичної моделі ймовірнісного розподілу даних. Ця модель може використовуватися для прогнозування майбутніх даних, аналізу даних або для отримання інформації про дані. Статистичне моделювання є унікальним методом моделювання, оскільки воно дозволяє враховувати випадковість. Це означає, що статистичні моделі можуть бути використані для моделювання систем, які є занадто складними або непередбачуваними для інших методів моделювання [3].

З переваг статистичного моделювання варто зазначити, що воно враховує випадковість, має широкий спектр застосувань та статистичні моделі можна тестувати на основі даних, які не використовувались для навчання. Але статистичне моделювання також має і недоліки: воно залежить від даних, статистичні моделі не можуть гарантувати точність своїх прогнозів і вони можуть бути складними для інтерпретації.

## 1.2 Ціль моделювання

На регулювальну ділянку цеху через випадкові інтервали часу надходять по два агрегати в середньому через кожні 30 хвилин. Первинне регулювання здійснюється для двох агрегатів одночасно і займає біля 30 хвилин. Якщо в момент приходу агрегатів попередня партія не була оброблена, агрегати на регулювання не приймаються. Агрегати, які одержали відмову, після первинного регулювання надходять у проміжний накопичувач. З накопичувача агрегати, що пройшли первинне регулювання, надходять попарно на вторинне регулювання, яке виконується в середньому за 30 хвилин, а ті, що не пройшли первинне регулювання, надходять на повне регулювання, що займає 100 хвилин для одного агрегату.

Головною ціллю є визначити ймовірність відмови в первинному регулюванні і завантаження накопичувача агрегатами, що потребують повного регулювання.

Також потрібно визначити параметри і розмір накопичувача, щоб було забезпечено безвідмовне обслуговування агрегатів, що надходять.

## 1.3 Концептуальна модель

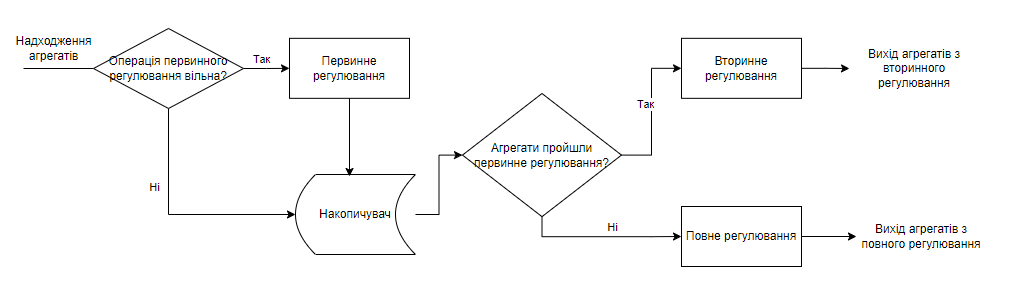


Рисунок 1.1 – Концептуальна модель

## 1.4 Вхідні та вихідні дані, параметри моделі

Таблиця 1.1 – Вхідні параметри моделі

|  |  |  |
| --- | --- | --- |
| **Параметри** | **Значення** | **Опис** |
| Т1 | 30(exp) | Періодичність поступання агрегатів |
| T2 | 30(exp) | Час здійснення первинного регулювання |
| T3 | 30(exp) | Час здійснення вторинного регулювання |
| T4 | 100(exp) | Час здійснення повного регулювання |
| N1 | 2 | Кількість агрегатів, що поступають до регулювальної ділянки кожні Т1 одиниць часу |
| N2 | 2 | Кількість агрегатів, що поступають до первинного регулювання і обробляються Т2 одиниць часу |
| N3 | 2 | Кількість агрегатів, що поступають до вторинного регулювання і обробляються Т3 одиниць часу |
| N4 | 1 | Кількість агрегатів, що обробляються Т4 одиниць часу на повному регулюванні. |
| Q | 1440 | Час моделювання |

Таблиця 1.2 – Вихідні параметри моделі

|  |  |
| --- | --- |
| **Параметри** | **Опис** |
| Q1 mean | Середня довжина черги оброблених первинним регулюванням у накопичувачі |
| Q1 max | Максимальна довжина черги оброблених первинним регулюванням у накопичувачі |
| Q2 mean | Середня довжина черги необроблених первинним регулюванням у накопичувачі |
| Q2 max | Максимальна довжина черги необроблених первинним регулюванням у накопичувачі |
| M1 | Кількість агрегатів з вторинного регулювання |
| M2 | Кількість агрегатів з повного регулювання |
| P | Ймовірність відмови в первинному регулюванні |

## 1.5 Обмеження

Для системи існують наступні обмеження:

* N1 > 0
* N2 > 0
* N3> 0
* N4> 0
* T1> 0
* T2> 0
* T3 > 0
* T4 > 0

## 1.6 Цільова функція

В нашому випадку ціллю моделювання є визначення ймовірності відмови в первинному регулюванні та завантаження накопичувача агрегатами. Таким чином цільова функція матиме вигляд:

Інша ціль нашого моделювання – це визначення оптимального розміру накопичувача, при якому буде забезпечено безвідмовне обслуговування агрегатів. Для цього дізнаємось ймовірність того, що агрегат залишиться в накопичувачі та теоретично виміряємо якого розміру має бути накопичувач для безвідмовної роботи регулювальної ділянки.

# РОЗДІЛ 2. ФОРМАЛІЗОВАНА МОДЕЛЬ

## 2.1 Елементи мережі Петрі

Після того, як ми створили концептуальну модель, нам потрібно визначити, як її формалізувати. Для цього існує два основних способи: мережі Петрі та мережі масового обслуговування.

Мережі Петрі - це потужний інструмент для моделювання дискретних процесів, які мають складні взаємозв'язки. Вони використовуються, коли потрібно моделювати систему з спільними ресурсами, які обслуговують багато процесів, або коли потрібно моделювати паралельні процеси. Важливою перевагою мереж Петрі є їхня універсальність, яка дозволяє моделювати як процеси управління, так і функціонування об'єктів управління [6].

На зображенні 2.1[4] показані компоненти формалізації мережі Петрі.



Рисунок 2.1 – Елементи мережі Петрі

## 2.2 Побудова формалізованої моделі

Генератор приблизно кожні 30 хвилин створює по два агрегати і після цього, агрегати потрапляють у позицію P2. Якщо первинне регулювання вільне, то агрегати попарно обробляються приблизно 30 хвилин і після цього потрапляють до основного накопичувача. Якщо первинне регулювання зайняте, агрегати потрапляють в накопичувач. З накопичувача агрегати які обробились потраплять у вторинне регулювання, якщо воно вільне вони обробляються приблизно 30 хвилин, а якщо ні, вони просто чекають обробки. З накопичувача агрегати які не обробились потрапляють у повне регулювання, якщо воно вільне один агрегат обробляється 100 хвилин, а якщо зайнятий, агрегати просто чекають обробки. Одиницею модельного часу є 1 хвилина.

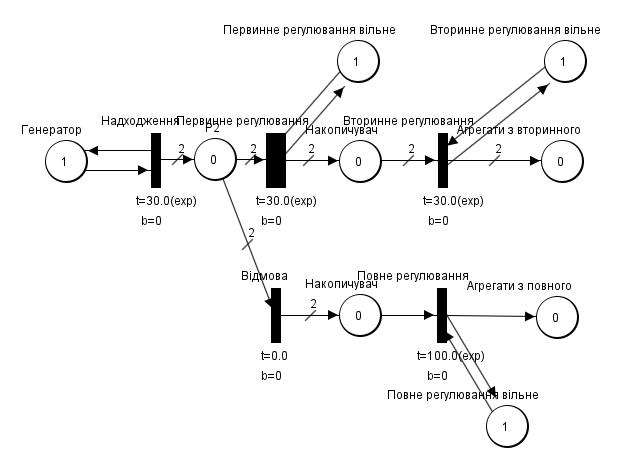


Рисунок 2.2 – Схема мережі Петрі, що відповідає моделі

## 2.3 Обчислення вихідних характеристики

Визначення середньої довжини черги агрегатів у основному накопичувачі:

(2.1),

Де черга агрегатів у основному накопичувачіk – k-те спостереження черги, Tmod – час моделювання, – зміна часу під час k-того спостереження

Визначення середньої довжини черги агрегатів у допоміжному накопичувачі:

(2.2),

Де черга агрегатів у допоміжному накопичувачі 2k – k-те спостереження черги, Tmod – час моделювання, – зміна часу під час k-того спостереження

Визначення ймовірності пропуску секції:

(2.3),

Де – Кількість пропусків секції, – Кількість заповнених секцій

# РОЗДІЛ 3. РЕАЛІЗАЦІЯ МОДЕЛІ

Для програмної реалізації даного підходу було обрано програмне забезпечення PetriObjectModelPaint. Це програмне забезпечення має вбудовані засоби генерації випадкових величин за експоненційним, рівномірним і нормальним законами розподілу. Крім того, до коду програми можна додавати додаткові спеціальні класи та методи, що дозволяє підлаштувати програмну реалізацію конкретної задачі та розробити функціонал для підрахунку статистичних даних, виведення звітів у зручному форматі та автоматизованого багаторазового проведення експериментів з ітеративним зміненням параметрів моделювання.

Програмне забезпечення має графічний інтерфейс, що дозволяє створити та візуалізувати модель. Можливість збереження візуального представлення як методу (у вигляді програмного коду) значно спрощує програмну реалізацію поставленої задачі. Крім того, програмне забезпечення дозволяє зберегти модель у вигляді файлу, що дозволяє проаналізувати модель багаторазово, не створюючи саму модель заново.

Програмне забезпечення для моделювання Петрі-об'єктів розроблено на мові Java. Воно складається з пакету PetriObjLib, який реалізує алгоритм симуляції Петрі-об'єктів, і пакетів, що забезпечують графічне представлення мережі. Якщо мережа Петрі містить перехід без вхідних або вихідних місць, виникне помилка.

Після підготовки списку Петрі-об'єктів та визначення зв'язків між ними модель може бути створена за допомогою класу PetriObjModel. Метод go(double time) цього класу ініціює моделювання. Якщо генератор затримки поверне від'ємне значення, виникне помилка.

Основними завданнями програмного забезпечення є забезпечення правильного алгоритму моделювання та правильних результатів, включаючи середні значення маркерів у місцях мережі Петрі, середнє значення буферів у переходах та стан мережі Петрі в кінці моделювання.

Для вирішення задачі були розроблені додаткові основні методи, які наведені у Додатку А. Таблиця 3.1 – Основні методи:

|  |  |
| --- | --- |
| Назва функції | Опис |
| CreateAdjSectionNet | Метод, що створює об’єкту регулювальної ділянки цеху |
| getModel | Метод, що збирає Петрі-об`єкти в модель і виконує прив`язку |
| showStatistics | Метод, який проводить експерименти і виводить статистику з отриманих даних. Також метод включає у собі цільові функції та дисперсійний аналіз |
| validateModel | Метод, в якому ми порівнюємо теоретичний і практичний час надходження та обробок |
| chebishevExperiment | Метод, в якому ми знаходимо достатню кількість експериментів для подальшого аналізу |
| timeExperiment | Метод, в якому ми проводимо експеримент з часом для майбутньої побудови графіку залежності |
| firstExperiment | Метод, в якому модель спрацьовує один раз і виводить статистику з отримних даних |

## 3.1 Опис програмної реалізації імітаційної моделі

Проведемо пробний експеримент завдяки створеному раніше методу firstExperiment з наступними вхідними даними:

Таблиця 3.2 – Вхідні дані у пробному експерименті

|  |  |
| --- | --- |
| Параметри | Значення |
| Т1 | 30(exp) |
| T2 | 30(exp) |
| T3 | 30(exp) |
| T4 | 100(exp) |
| N1 | 2 |
| N2 | 2 |
| N3 | 2 |
| N4 | 1 |
| Q | 1440 |

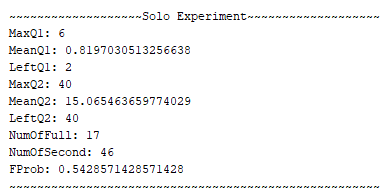


Рисунок 3.1 – Результат пробного експерименту

Як можемо побачити, за час моделювання в один день (1440 хвилин) 46 агрегатів пройшли вторинне регулювання, 17 агрегатів пройшли повне регулювання, 40 агрегатів залишись у накопичувачі. Відсоток того, що первинне регулювання зайняте у час надходження становить 54%.

## 3.2 Оцінка адекватності моделі

Використаємо створений раніше метод validateModel, який порівнює теоретичні та практичні значення часу надходження, часу обробки первинним регулюванням, вторинним регулюванням і повним регулюванням.

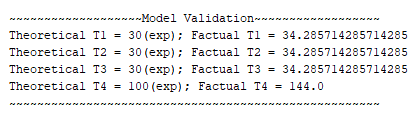


Рисунок 3.2 – Результат оцінки адекватності моделі

Виконаємо пробний експеримент з минулого пункту ще раз, щоб оцінити модель на адекватність.

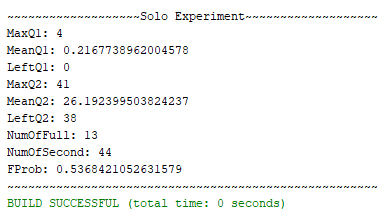


Рисунок 3.3 – Результат пробного експерименту

Як можемо побачити, за 1440 хвилин 44 агрегати пройшли вторинну обробку, 13 агрегатів пройшло повну обробку та 38 агрегатів залишились у накопучивачі, тобто усього 95 агрегатів надійшло на регулювальну ділянку цеху.

Перевіримо час надходження агрегатів з часом 30 хвилин:

(3.1)

Як можемо побачити, практичне та теоретичні значення співпадають. Також обрахуємо приблизний час надходження агрегатів:

(3.2)

Також зробимо це для обробки агрегатів у кожному регулюванні. Для вторинного регулювання:

(3.3)

Для первинного регулювання:

(3.4)

Для повного регулювання:

(3.5)

Як можемо побачити, теоретичні значення співпадають з практичними, тому можемо зробити висновок, що програма функціонує правильно. Отже, модель є адекватною.

## 3.3 Верифікація моделі

Зробимо верифікацію моделі завдяки методу showStatistics. Для цього ми створимо кілька різних наборів вхідних значень і запустимо кожен з них 5 разів і порівняємо результати.

Таблиця 3.3 – Набори вхідних значень для експериментів

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Час надходження | Час обробки пер.рег. | Час обробки втор.рег. | Час обробки пов. рег. | К-сть ресурсу  пер. рег. | К-сть ресурсу  втор. рег. | К-сть ресурсу  пов. рег. |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 |
| 20 | 20 | 20 | 80 | 1 | 3 | 2 |

Для генерації верифікаційної таблиці, проведемо експерименти поступово змінюючи параметри часу надходження, обробки та розмір вільного ресурсу у первинному, вторинному та повному регулюванні.

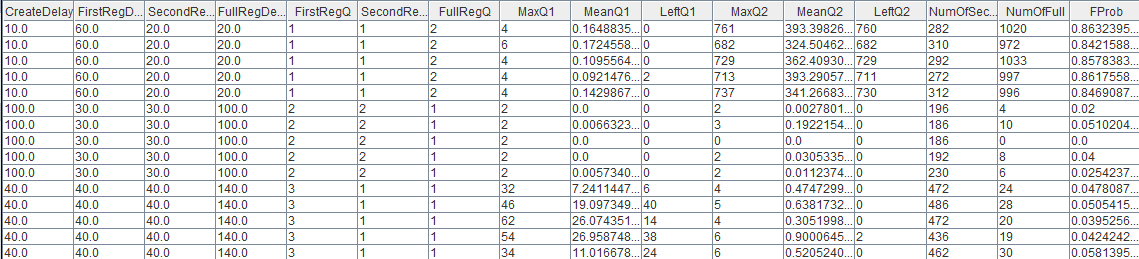


Рисунок 3.5 – Верифікація моделі

Як можемо побачити, якщо час первинного регулювання більший за час надходження, то відсоток відмови буде більшим, а коли час надходження більший за час первинного регулювання, відсоток буде майже нульовим. Також при збільшенні ресурсів, при однаковому часі, відсоток також буде нульовим, адже агрегати майже і не будуть потрапляти до повного регулювання.

Це відповідає очікуванням, тому модель є адекватною.

# РОЗДІЛ 4. ПРОВЕДЕННЯ ЕКСПЕРИМЕНТІВ НА МОДЕЛІ

## 4.1 Постановка задачі

Основна ціль моделювання даної системи є визначення відсотку відмови первинного регулювання, якщо агрегати прийшли у момент, коли регулювання вже зайняте. Також необхідно визначити приблизний розмір накопичувача, щоб програма працювала безвідмовно.

Значення цільової функції – мінімізація ймовірності пропуску первинного регулювання, а також розрахування приблизного розміру накопичувача. Експерименти будемо проводити послідовно, задаючи різні вхідні дані.

## 4.2 Тактичне планування

Проведемо пробний прогін завдяки методу timeExperiment та побудуємо графік залежності ймовірності пропуску від часу моделювання.

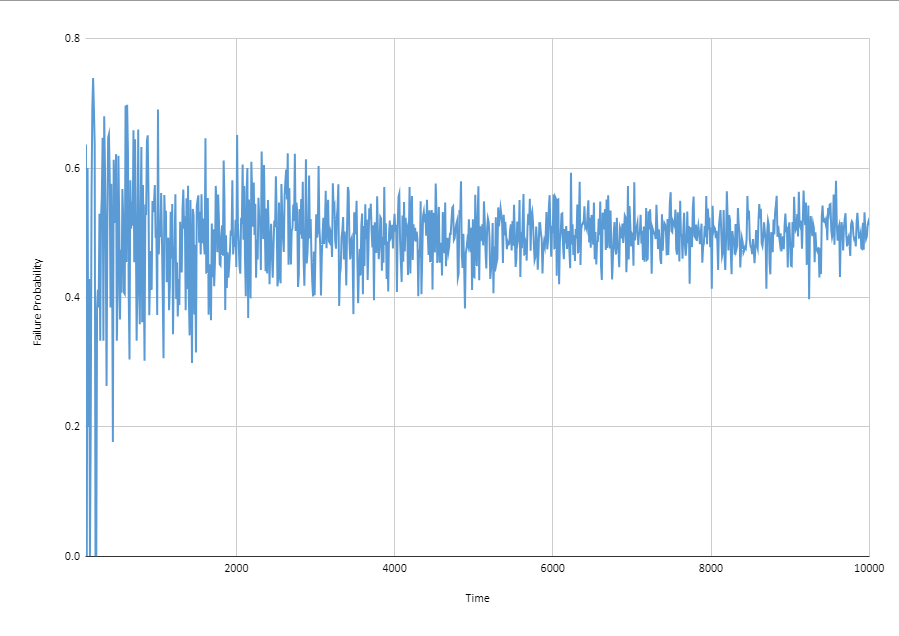


Рисунок 4.1 – Графік залежності ймовірності пропуску секції від часу моделювання

На рисунку 4.1 можемо побачити, що в цілому час моделювання не сильно впливає на роботу моделі, але після приблизно 4000 од.часу графік більш стабільний та коливається у діапазоні від 40 до 60 відсотків. Тому візьмемо тривалість прогону в 4000 одиниць модельного часу для наступних експериментів, щоб перехідні процеси мало впливали на результат. Для визначення кількості необхідних експериментів за нерівністю Чебишева, скористаємося формулою:

(4.1)

N – це кількість прогонів, ε – точність оцінки (5% від середнього значення), - довірча ймовірність (0.95), σ – дисперсія.

Скористаємось розробленим раніше методом chebishevExperiment для пошуку достатньої для заданої точності кількості експериментів відповідно до формули (4.1).

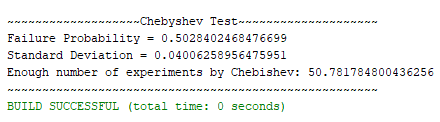


Рисунок 4.2 – Результат пошуку мінімальної кількості експериментів

## 4.3 Стратегічне планування

Отже, для заданих параметрів потрібно провести 50 запусків. Далі проведемо тестування для всіх можливих комбінацій часу надходження, обробки та вільного ресурсу у пунктах регулювань. Для кожного з цих варіантів також проведемо по 50 запусків. Результати цих запусків наведені в додатку Б.

На підставі отриманих даних з додатку Б проведемо аналіз дисперсії. Суть дисперсійного аналізу полягає в тому, що ми порівнюємо дисперсію, яка виникає внаслідок впливу факторів, з дисперсією, яка пояснюється випадковими причинами. Якщо різниця між цими видами дисперсії велика, то фактори справді впливають на відгук моделі.

Спочатку, обрахуємо середні значення за формулами[4]:

(4.2),

Де p – кількість прогонів у кожному експерименті, q – рівень фактора, а – ймовірність пропуску секції j-ого експерименту в i-тому спостереженні

Далі введемо величини[4]:

(4.3)

Наступним кроком буде визначити загальну, факторну та залишкову дисперсії[4]:

(4.4)

Далі знаходимо критерій Фішера[4]:

(4.5)

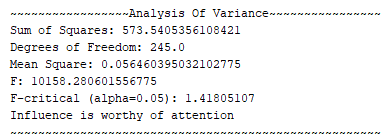


Рисунок 4.3 – Отримані результати проведення дисперсійного аналізу

Як можемо побачити, критерій Фішера 10158.281 при критичному значенні 1.418, а це означає, що вплив на режим роботи комплектуючого конвеєра є значним.

Тепер використаємо отримані дані для того щоб порахувати оптимальний розмір накопичувача, щоб регулювальна ділянка працювала безвідмовно. Для цього ми порахуємо кількість агрегатів які пройшли вторинне та повне регулювання та максимальну кількість агрегатів які були у накопичувачі. Завдяки цьому ми порахуємо відсоток на те, що агрегати залишаться у накопичувачі.

Для того, щоб віднайти оптимальний розмір накопичувача, я порахував теоретичний час надходження за формулою (3.1) і помножив його на знайдений раніше відсоток. Після цього, результат було округлено.

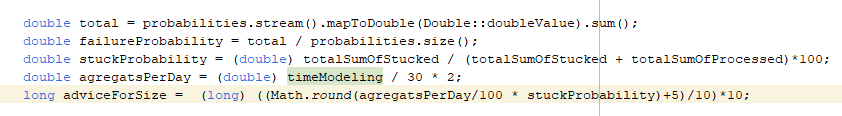


Рисунок 4.4 – Пошук оптимального розміру накопичувача

Скористаємось створеним раніше методом showStatistics для того, щоб порахувати оптимальний розмір накопичувача.

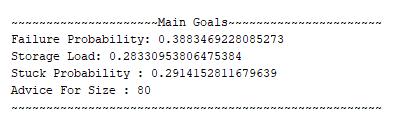


Рисунок 4.5 – Результат пошуку оптимального розміру накопичувача при часу моделювання 4000хв

Як можемо побачити, якщо регулювальна ділянка буде працювати 4000 хвилин, відсоток агрегатів, що не дійшли до кінця буде становити 28.7%, а також рекомендовано мати накопичувач з розміром 80. Але якщо регулювальна ділянка буде працювати більше або менше, цей розмір відповідно буде змінюватись. Також ми розрахували завантаження накопичувача, яке становить 29.14%.

# РОЗДІЛ 5. ІНТЕРПРЕТАЦІЯ РЕЗУЛЬТАТІВ МОДЕЛЮВАННЯ

У дослідженні було проведено перевірку моделі, розраховано приблизний час виконання процесів у системі, побудовано графік залежності ймовірності пропуску секції від часу моделювання, визначено оптимальну кількість експериментів та проведено дисперсійний аналіз. Це дозволило краще зрозуміти залежність ймовірності пропуску первинного регулювання від вхідних параметрів системи.

Цільові завдання дослідження – визначити ймовірність відмови в первинному регулюванні та завантаження накопичувача агрегатами, що потребують повного регулювання. Визначити параметри та ввести в систему накопичувач, що забезпечує безвідмовне обслуговування агрегатів, що надходять.

Ймовірність відмови становить приблизно 30-50%. Для її зменшення рекомендовано забирати більше агрегатів на первинне регулювання або приймати менше агрегатів. Для зменшення завантаження на накопичувач рекомендовано пришвидшити час обробки повним регулюванням або забрати більше агрегатів. Відсоток, який залишається у накопичувачі, становить приблизно 30-40%.

Рекомендований розмір накопичувача для безвідмовної роботи ділянки залежить від часу роботи. При часі роботи в 1440 хвилин, що дорівнює одному дню, рекомендовано поставити накопичувач, який вміщає 40 агрегатів. При часі роботи в 4000 хвилин, цей розмір збільшується до 80. Для оптимізації розміру необхідно пришвидшити час обробки вторинним та повним регулюванням або додавати додаткові ресурси, тобто обробляти більше агрегатів.

# ВИСНОВОК

У цій роботі розглянута проблема визначення оптимального режиму роботи регулювальної ділянки цеху, який забезпечує мінімальну ймовірність того, що обладнання не буде відрегульоване вчасно. Було досліджено можливі методи розв'язання цієї проблеми, розроблено концептуальну та формалізовану моделі, виконано програмну реалізацію моделі та розв'язано поставлену задачу. Також проведено аналіз експериментально отриманих даних.

Для розв'язання поставленої задачі було використано мову програмування Java, бібліотеку PetriObjModelPaint та розроблену імітаційну модель. Простота зміни вхідних даних та результати моделювання дозволяють дослідити роботу регулювальної ділянки.

Відповідно до проведених експериментів, ми можемо сказати, що рекомендований розмір накопичувача за один день дорівнюватиме 30. Також відсоток відмови дорівнює 30-50%, але його можна покращити, якщо пришвидшити первинну обробку, або виділити на неї додаткові ресурси. При тестуванні ми перевірили, що модель є адекватною

.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Основні види моделювання. Формальні методи побудови моделей ПНС ХНЕУ ім. С. Кузнеця – 2017. – 18 с.
2. Імітаційне моделювання [Електронний ресурс] – Режим доступу до ресурсу: <https://stud.com.ua/98833/informatika/imitatsiyne_modelyuvannya>
3. An Introduction to Statistical Learning / Daniela Witten, Gareth M. James, Trevor Hastie, Robert Tibshirani. – 2013 – 440 c.
4. Стеценко І.В. Моделювання систем: Навчальний посібник / І.В. Стеценко; М-во освіти і науки України, Черк. держ. технол. ун-т. – Черкаси: ЧДТУ, 2011. – 407с.
5. Імітаційне моделювання систем та процесів: Електронне навчальне видання. Конспект лекцій / В.Б. Неруш, В.В. Курдеча. – К.: НН ІТС НТУУ «КПІ», 2012. – 115 с.
6. Веб-сервіс моделювання дискретно-подійних систем / Дифучин А.Ю; КПІ ім. Ігоря Сікорського – Київ, 2018. – 95 с.
7. Стеценко І. В. бібліотека «PetriObjModelPaint» - URL: <https://github.com/StetsenkoInna/PetriObjModelPaint>

# Додаток А. Лістинг коду

package LibNet;

import PetriObj.PetriObjModel;

import PetriObj.PetriSim;

import PetriObj.ArcIn;

import PetriObj.ArcOut;

import PetriObj.ExceptionInvalidNetStructure;

import PetriObj.ExceptionInvalidTimeDelay;

import PetriObj.PetriNet;

import PetriObj.PetriP;

import PetriObj.PetriT;

import java.util.ArrayList;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTable;

/\*\*

\*

\* @author white

\*/

public class coursework1 {

public static void main(String[] args) throws ExceptionInvalidTimeDelay, ExceptionInvalidNetStructure {

int timeModeling = 1440;

int numOfExp = 50;

//firstExperiment(timeModeling);

showStatistics(numOfExp, timeModeling);

//validateModel(timeModeling);

//chebishevExperiment(timeModeling);

//timeExperiment(numOfExp);

}

public static void firstExperiment(int timeModeling)throws

ExceptionInvalidTimeDelay, ExceptionInvalidNetStructure {

PetriObjModel model = getModel(30, 30, 1, 30, 1, 100, 1);

model.setIsProtokol(false);

model.go(timeModeling);

double failureProbability = (model.getListObj().get(0).getNet().getListP()[6].getMark() + model.getListObj().get(0).getNet().getListP()[4].getMark())

/ ((double)(model.getListObj().get(0).getNet().getListP()[6].getMark() + model.getListObj().get(0).getNet().getListP()[4].getMark()

+ model.getListObj().get(0).getNet().getListP()[3].getMark()) + model.getListObj().get(0).getNet().getListP()[2].getMark());

System.out.println();

System.out.println("~~~~~~~~~~~~~~~~~~~Solo Experiment~~~~~~~~~~~~~~~~~~~");

System.out.println("MaxQ1: " + model.getListObj().get(0).getNet().getListP()[2].getObservedMax());

System.out.println("MeanQ1: " + model.getListObj().get(0).getNet().getListP()[2].getMean());

System.out.println("LeftQ1: " + model.getListObj().get(0).getNet().getListP()[2].getMark());

System.out.println("MaxQ2: " + model.getListObj().get(0).getNet().getListP()[4].getObservedMax());

System.out.println("MeanQ2: " + model.getListObj().get(0).getNet().getListP()[4].getMean());

System.out.println("LeftQ2: " + model.getListObj().get(0).getNet().getListP()[4].getMark());

System.out.println("NumOfFull: " + model.getListObj().get(0).getNet().getListP()[6].getMark());

System.out.println("NumOfSecond: " + model.getListObj().get(0).getNet().getListP()[3].getMark());

System.out.println("FProb: " + failureProbability);

System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

}

public static void showStatistics(int numOfExp, int timeModeling) throws

ExceptionInvalidTimeDelay, ExceptionInvalidNetStructure {

JFrame frame = new JFrame();

String[] columnNames = {"CreateDelay", "FirstRegDelay", "SecondRegDelay", "FullRegDelay","FirstRegQ", "SecondRegQ", "FullRegQ", "MaxQ1" ,"MeanQ1" ,"LeftQ1" ,"MaxQ2" ,"MeanQ2" ,"LeftQ2", "NumOfSecond" ,"NumOfFull", "FProb"};

double[] createDelay = {10.0, 100.0, 40.0, 30.0, 20.0};

double[] firstRegDelay = {60.0, 30.0, 40.0, 30.0, 20.0};

int[] firstRegQ = {1, 2, 3, 1, 1};

double[] secondRegDelay = {20.0, 30.0, 40.0, 30.0, 50.0};

int[] secondRegQ = {1, 2, 1, 1, 3};

double[] fullRegDelay = {20.0, 100.0, 140.0, 100.0, 80.0, };

int[] fullRegQ = {2, 1, 1, 1, 2};

int numOfSituations = createDelay.length;

Object[][] data = new Object[numOfSituations\*numOfExp][columnNames.length];

int index = 0;

ArrayList<Double> probabilities = new ArrayList<>();

ArrayList<Double> groupDeviation = new ArrayList<>();

ArrayList<Double> probabilitiesAvg = new ArrayList<>();

ArrayList<Double> difference = new ArrayList<>();

int totalSumOfProcessed = 0;

int totalSumOfStucked = 0;

for(int i = 0; i < numOfSituations; i++){

for(int j = 0; j < numOfExp; j++){

PetriObjModel model = getModel(createDelay[i], firstRegDelay[i], firstRegQ[i], secondRegDelay[i], secondRegQ[i], fullRegDelay[i], fullRegQ[i]);

model.setIsProtokol(false);

model.go(timeModeling);

double failureProbability = (model.getListObj().get(0).getNet().getListP()[6].getMark() + model.getListObj().get(0).getNet().getListP()[4].getMark())

/ ((double)(model.getListObj().get(0).getNet().getListP()[6].getMark() + model.getListObj().get(0).getNet().getListP()[4].getMark()

+ model.getListObj().get(0).getNet().getListP()[3].getMark()) + model.getListObj().get(0).getNet().getListP()[2].getMark());

totalSumOfProcessed = totalSumOfProcessed + model.getListObj().get(0).getNet().getListP()[3].getMark()+ model.getListObj().get(0).getNet().getListP()[6].getObservedMax();

totalSumOfStucked = totalSumOfStucked + model.getListObj().get(0).getNet().getListP()[2].getMark() + model.getListObj().get(0).getNet().getListP()[4].getObservedMax();

data[index] = new Object[]{createDelay[i], firstRegDelay[i], secondRegDelay[i], fullRegDelay[i], firstRegQ[i], secondRegQ[i], fullRegQ[i],

model.getListObj().get(0).getNet().getListP()[2].getObservedMax(),

model.getListObj().get(0).getNet().getListP()[2].getMean(),

model.getListObj().get(0).getNet().getListP()[2].getMark(),

model.getListObj().get(0).getNet().getListP()[4].getObservedMax(),

model.getListObj().get(0).getNet().getListP()[4].getMean(),

model.getListObj().get(0).getNet().getListP()[4].getMark(),

model.getListObj().get(0).getNet().getListP()[3].getMark(),

model.getListObj().get(0).getNet().getListP()[6].getMark(),

failureProbability

};

index++;

probabilities.add(failureProbability);

}

double total = probabilities.stream().mapToDouble(Double::doubleValue).sum();

double failureProbability = total / probabilities.size();

for(int j = 0; j<numOfExp; j++){

probabilitiesAvg.add(failureProbability);

}

}

JTable table = new JTable(data, columnNames);

JScrollPane sp = new JScrollPane(table);

frame.add(sp);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(1920,1080);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

double total = probabilities.stream().mapToDouble(Double::doubleValue).sum();

double failureProbability = total / probabilities.size();

double stuckProbability = (double) totalSumOfStucked / (totalSumOfStucked + totalSumOfProcessed)\*100;

double agregatsPerDay = (double) timeModeling / 30 \* 2;

long adviceForSize = (long) ((Math.round(agregatsPerDay/100 \* stuckProbability)+5)/10)\*10;

System.out.println();

System.out.println("~~~~~~~~~~~~~~~~~~~~~Main Goals~~~~~~~~~~~~~~~~~~~~~~");

System.out.println("Failure Probability: " + failureProbability);

System.out.println("Stack Probability : " + stuckProbability);

System.out.println("Advice For Size : " + adviceForSize);

System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

for (int i = 0; i < probabilities.size(); i++) {

groupDeviation.add(Math.pow((probabilities.get(i) - probabilitiesAvg.get(i)), 2));

difference.add(Math.pow((probabilitiesAvg.get(i) - failureProbability), 2));

}

double sFactual = difference.stream().mapToDouble(Double::doubleValue).sum() \* numOfExp;

double sResidual = groupDeviation.stream().mapToDouble(Double::doubleValue).sum();

double degreesOfFreedom = numOfSituations \* (numOfExp - 1);

double dFactual = sFactual;

double dResidual = sResidual / degreesOfFreedom;

double f = dFactual / dResidual;

double fCritical = 3.01; //a=0.05; k1=4; k2=16

System.out.println();

System.out.println("~~~~~~~~~~~~~~~~~Analysis Of Variance~~~~~~~~~~~~~~~~");

System.out.println("Sum of Squares: " + dFactual);

System.out.println("Degrees of Freedom: " + degreesOfFreedom);

System.out.println("Mean Square: " + dResidual);

System.out.println("F: " + f);

System.out.println("F-critical (alpha=0.05): " + fCritical);

if (f > fCritical) {

System.out.println("Influence is worthy of attention");

} else {

System.out.println("Influence is not worthy of attention");

}

System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

}

public static void validateModel(int timeModeling) throws ExceptionInvalidTimeDelay, ExceptionInvalidNetStructure {

PetriObjModel model = getModel(30, 30, 1, 30, 1, 100, 1);

model.setIsProtokol(false);

model.go(timeModeling);

System.out.println();

System.out.println("~~~~~~~~~~~~~~~~~~~Model Validation~~~~~~~~~~~~~~~~~~");

double practicalT1 = timeModeling / (double)((model.getListObj().get(0).getNet().getListP()[3].getMark() + model.getListObj().get(0).getNet().getListP()[6].getMark()+ model.getListObj().get(0).getNet().getListP()[2].getMark() + model.getListObj().get(0).getNet().getListP()[4].getMark())/2);

System.out.println("Theoretical T1 = 30(exp); Factual T1 = " + practicalT1);

double practicalT2 = timeModeling / (double)(model.getListObj().get(0).getNet().getListP()[2].getMark()+ model.getListObj().get(0).getNet().getListP()[3].getMark());

System.out.println("Theoretical T2 = 30(exp); Factual T2 = "+ practicalT2);

double practicalT3 = timeModeling / (double)((model.getListObj().get(0).getNet().getListP()[3].getMark()));

System.out.println("Theoretical T3 = 30(exp); Factual T3 = " + practicalT3);

double practicalT4 = timeModeling / (double)((model.getListObj().get(0).getNet().getListP()[6].getMark()));

System.out.println("Theoretical T4 = 100(exp); Factual T4 = " + practicalT4);

System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

}

public static void chebishevExperiment(int timeModeling) throws ExceptionInvalidTimeDelay, ExceptionInvalidNetStructure {

int runAmount = 100;

ArrayList<Double> probabilities = new ArrayList<>();

for (int i = 0; i < runAmount; i++) {

PetriObjModel model = getModel(30, 30, 1, 30, 1, 100, 1);

model.setIsProtokol(false);

model.go(timeModeling);

double failureProbability = (model.getListObj().get(0).getNet().getListP()[6].getMark() + model.getListObj().get(0).getNet().getListP()[4].getMark())

/ ((double)(model.getListObj().get(0).getNet().getListP()[6].getMark() + model.getListObj().get(0).getNet().getListP()[4].getMark()

+ model.getListObj().get(0).getNet().getListP()[3].getMark()) + model.getListObj().get(0).getNet().getListP()[2].getMark());

probabilities.add(failureProbability);

}

double total = probabilities.stream().mapToDouble(Double::doubleValue).sum();

double avg = total / probabilities.size();

double sum = 0;

for (Double finProb : probabilities) {

sum += Math.pow((finProb - avg), 2);

}

double stdDev = Math.sqrt(sum / (probabilities.size()-1));

double numOfExp = Math.pow(stdDev, 2) / (Math.pow((0.05 \* avg), 2) \* (1 - 0.95));

System.out.println();

System.out.println("~~~~~~~~~~~~~~~~~~~Chebyshev Test~~~~~~~~~~~~~~~~~~~~");

System.out.println("Failure Probability = " + avg);

System.out.println("Standard Deviation = " + stdDev);

System.out.println("Enough number of experiments by Chebishev: " + numOfExp);

System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

}

public static void timeExperiment(int numOfExp) throws ExceptionInvalidTimeDelay, ExceptionInvalidNetStructure {

int timeModelingMax = 10000;

int timeModelingStep = 10;

int timeModelingStart = 100;

JFrame frame = new JFrame();

Object[][] data = new Object[(timeModelingMax-timeModelingStep)\*numOfExp][2];

String[] columnNames = {"CreateDelay", "FirstRegDelay"};

int index = 0;

for (int i = 0; i < numOfExp; i++) {

int t = timeModelingStart;

while (t <= timeModelingMax) {

PetriObjModel model = getModel(30, 30, 1, 30, 1, 100, 1);

model.setIsProtokol(false);

model.go(t);

double failureProbability = (model.getListObj().get(0).getNet().getListP()[6].getMark() + model.getListObj().get(0).getNet().getListP()[4].getMark())

/ ((double)(model.getListObj().get(0).getNet().getListP()[6].getMark() + model.getListObj().get(0).getNet().getListP()[4].getMark()

+ model.getListObj().get(0).getNet().getListP()[3].getMark()) + model.getListObj().get(0).getNet().getListP()[2].getMark());

data[index] = new Object[]{t, failureProbability};

t += timeModelingStep;

index++;

}

}

JTable table = new JTable(data, columnNames);

JScrollPane sp = new JScrollPane(table);

frame.add(sp);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(1920,1080);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

public static PetriObjModel getModel(double createDelay, double firstRegDelay, int firstRegQ, double secondRegDelay, int secondRegQ, double fullRegDelay, int fullRegQ) throws ExceptionInvalidTimeDelay, ExceptionInvalidNetStructure {

ArrayList<PetriSim> list = new ArrayList<>();

list.add(new PetriSim(CreateAdjSectionNet(createDelay, firstRegDelay, firstRegQ, secondRegDelay, secondRegQ, fullRegDelay, fullRegQ)));

return new PetriObjModel(list);

}

public static PetriNet CreateAdjSectionNet(double createDelay, double firstRegDelay, int firstRegQ, double secondRegDelay, int secondRegQ, double fullRegDelay, int fullRegQ) throws ExceptionInvalidNetStructure, ExceptionInvalidTimeDelay {

ArrayList<PetriP> d\_P = new ArrayList<>();

ArrayList<PetriT> d\_T = new ArrayList<>();

ArrayList<ArcIn> d\_In = new ArrayList<>();

ArrayList<ArcOut> d\_Out = new ArrayList<>();

d\_P.add(new PetriP("Генератор",1));

d\_P.add(new PetriP("P2",0));

d\_P.add(new PetriP("Накопичувач",0));

d\_P.add(new PetriP("Агрегати з вторинного",0));

d\_P.add(new PetriP("Накопичувач",0));

d\_P.add(new PetriP("Первинне регулювання вільне", firstRegQ ));

d\_P.add(new PetriP("Агрегати з повного",0));

d\_P.add(new PetriP("Вторинне регулювання вільне",secondRegQ));

d\_P.add(new PetriP("Повне регулювання вільне",fullRegQ));

d\_T.add(new PetriT("Надходження",createDelay));

d\_T.get(0).setDistribution("exp", d\_T.get(0).getTimeServ());

d\_T.get(0).setParamDeviation(0.0);

d\_T.add(new PetriT("Первинне регулювання",firstRegDelay));

d\_T.get(1).setDistribution("exp", d\_T.get(1).getTimeServ());

d\_T.get(1).setParamDeviation(0.0);

d\_T.get(1).setPriority(10);

d\_T.add(new PetriT("Вторинне регулювання",secondRegDelay));

d\_T.get(2).setDistribution("exp", d\_T.get(2).getTimeServ());

d\_T.get(2).setParamDeviation(0.0);

d\_T.add(new PetriT("Відмова",0.0));

d\_T.add(new PetriT("Повне регулювання", fullRegDelay));

d\_T.get(4).setDistribution("exp", d\_T.get(4).getTimeServ());

d\_T.get(4).setParamDeviation(0.0);

d\_In.add(new ArcIn(d\_P.get(1),d\_T.get(1),2));

d\_In.add(new ArcIn(d\_P.get(2),d\_T.get(2),2));

d\_In.add(new ArcIn(d\_P.get(1),d\_T.get(3),2));

d\_In.add(new ArcIn(d\_P.get(4),d\_T.get(4),1));

d\_In.add(new ArcIn(d\_P.get(7),d\_T.get(2),1));

d\_In.add(new ArcIn(d\_P.get(8),d\_T.get(4),1));

d\_In.add(new ArcIn(d\_P.get(5),d\_T.get(1),1));

d\_In.add(new ArcIn(d\_P.get(0),d\_T.get(0),1));

d\_Out.add(new ArcOut(d\_T.get(0),d\_P.get(1),2));

d\_Out.add(new ArcOut(d\_T.get(1),d\_P.get(2),2));

d\_Out.add(new ArcOut(d\_T.get(2),d\_P.get(3),2));

d\_Out.add(new ArcOut(d\_T.get(3),d\_P.get(4),2));

d\_Out.add(new ArcOut(d\_T.get(4),d\_P.get(6),1));

d\_Out.add(new ArcOut(d\_T.get(2),d\_P.get(7),1));

d\_Out.add(new ArcOut(d\_T.get(4),d\_P.get(8),1));

d\_Out.add(new ArcOut(d\_T.get(1),d\_P.get(5),1));

d\_Out.add(new ArcOut(d\_T.get(0),d\_P.get(0),1));

PetriNet d\_Net = new PetriNet("coursework\_v1.pns",d\_P,d\_T,d\_In,d\_Out);

PetriP.initNext();

PetriT.initNext();

ArcIn.initNext();

ArcOut.initNext();

return d\_Net;

}

}

# Додаток Б. Результати проведення експериментів

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CreateDelay** | **FirstRegDelay** | **SecondRegDelay** | **FullRegDelay** | **FirstRegQ** | **SecondRegQ** | **FullRegQ** | **MaxQ1** | **MeanQ1** | **LeftQ1** | **MaxQ2** | **MeanQ2** | **LeftQ2** | **NumOfSecond** | **NumOfFull** | **Fprob** |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.076206 | 0 | 310 | 157.9598 | 310 | 92 | 402 | 0.885572 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 6 | 0.288384 | 0 | 234 | 98.76342 | 230 | 120 | 406 | 0.84127 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.126913 | 0 | 214 | 128.2701 | 198 | 100 | 398 | 0.856322 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.02871 | 0 | 243 | 106.4817 | 234 | 122 | 398 | 0.838196 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 6 | 0.186226 | 0 | 291 | 111.5573 | 290 | 110 | 390 | 0.860759 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.124341 | 0 | 234 | 115.7821 | 233 | 112 | 411 | 0.851852 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.156693 | 0 | 284 | 149.1672 | 279 | 104 | 411 | 0.869018 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.003939 | 0 | 377 | 190.9561 | 377 | 110 | 383 | 0.873563 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.058093 | 0 | 322 | 164.0765 | 318 | 110 | 368 | 0.861809 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 6 | 0.181999 | 0 | 343 | 183.3453 | 338 | 96 | 392 | 0.883777 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.064303 | 0 | 305 | 143.6549 | 304 | 110 | 376 | 0.860759 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.128355 | 0 | 311 | 149.0974 | 310 | 116 | 418 | 0.862559 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.003781 | 0 | 358 | 186.0862 | 358 | 88 | 378 | 0.893204 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 6 | 0.146299 | 0 | 295 | 164.0269 | 294 | 108 | 382 | 0.862245 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.062214 | 0 | 261 | 127.1751 | 260 | 110 | 430 | 0.8625 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.004131 | 0 | 275 | 127.9312 | 275 | 98 | 413 | 0.875318 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.050231 | 0 | 271 | 130.2492 | 271 | 134 | 411 | 0.835784 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.134973 | 0 | 209 | 133.7639 | 205 | 114 | 411 | 0.843836 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 6 | 0.200524 | 0 | 250 | 120.1154 | 248 | 118 | 406 | 0.84715 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 8 | 0.296661 | 0 | 346 | 174.1696 | 342 | 114 | 400 | 0.866822 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 6 | 0.223072 | 0 | 239 | 115.8132 | 237 | 94 | 425 | 0.875661 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.138717 | 0 | 284 | 152.0934 | 284 | 128 | 380 | 0.838384 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.13842 | 0 | 275 | 117.7803 | 272 | 112 | 430 | 0.862408 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.093666 | 0 | 283 | 140.6435 | 276 | 110 | 428 | 0.864865 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.023945 | 0 | 300 | 154.1029 | 300 | 94 | 378 | 0.878238 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.30931 | 0 | 269 | 128.6058 | 265 | 106 | 405 | 0.863402 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.108412 | 0 | 216 | 118.9623 | 214 | 108 | 430 | 0.856383 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.2202 | 0 | 318 | 131.2121 | 315 | 124 | 401 | 0.852381 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.055493 | 0 | 375 | 192.2678 | 374 | 84 | 384 | 0.900238 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.107484 | 0 | 336 | 156.8332 | 333 | 132 | 369 | 0.841727 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.057052 | 0 | 291 | 138.4607 | 290 | 100 | 376 | 0.869452 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.04888 | 0 | 287 | 138.8528 | 287 | 110 | 395 | 0.861111 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 8 | 0.303365 | 0 | 285 | 142.208 | 283 | 114 | 381 | 0.85347 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 6 | 0.276582 | 0 | 336 | 155.9473 | 335 | 126 | 369 | 0.848193 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.16171 | 0 | 393 | 195.9121 | 392 | 122 | 348 | 0.858469 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 12 | 1.019091 | 0 | 241 | 98.56686 | 239 | 124 | 417 | 0.841026 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 8 | 0.358673 | 0 | 313 | 149.6007 | 313 | 116 | 381 | 0.85679 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 6 | 0.214287 | 0 | 257 | 130.9153 | 256 | 116 | 398 | 0.849351 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.106109 | 0 | 338 | 146.2765 | 338 | 106 | 384 | 0.871981 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.102324 | 0 | 327 | 148.5549 | 325 | 106 | 403 | 0.872902 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.125985 | 0 | 240 | 99.27656 | 238 | 116 | 392 | 0.844504 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 6 | 0.314822 | 0 | 248 | 133.077 | 246 | 132 | 448 | 0.840194 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 6 | 0.227526 | 0 | 335 | 140.0491 | 335 | 112 | 415 | 0.87007 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.042363 | 0 | 255 | 124.7135 | 252 | 112 | 404 | 0.854167 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.097108 | 0 | 287 | 152.3769 | 282 | 106 | 410 | 0.867168 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 2 | 0.065392 | 0 | 330 | 143.3685 | 330 | 106 | 394 | 0.872289 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.187231 | 0 | 345 | 188.2122 | 334 | 110 | 392 | 0.868421 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 4 | 0.077354 | 0 | 312 | 172.4466 | 312 | 114 | 390 | 0.860294 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 6 | 0.142418 | 0 | 305 | 154.9123 | 305 | 102 | 357 | 0.866492 |
| 10 | 60 | 20 | 20 | 1 | 1 | 2 | 6 | 0.153032 | 0 | 283 | 148.7668 | 279 | 114 | 419 | 0.859606 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.010293 | 0 | 3 | 0.073652 | 0 | 60 | 4 | 0.0625 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 74 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 2 | 0.01051 | 0 | 70 | 2 | 0.027778 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 3 | 0.06525 | 3 | 70 | 0 | 0.041096 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.00615 | 0 | 5 | 0.551739 | 0 | 92 | 6 | 0.061224 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 4 | 0.026439 | 0 | 3 | 0.142043 | 0 | 80 | 4 | 0.047619 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.029043 | 0 | 2 | 0.07267 | 0 | 72 | 4 | 0.052632 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.014877 | 0 | 3 | 0.338756 | 0 | 86 | 6 | 0.065217 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 2 | 0.009119 | 0 | 76 | 2 | 0.025641 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 76 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.008561 | 0 | 0 | 0 | 0 | 82 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.013288 | 0 | 3 | 0.091305 | 0 | 78 | 4 | 0.04878 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 64 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 2 | 0.012645 | 0 | 68 | 2 | 0.028571 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.010943 | 0 | 3 | 0.312371 | 0 | 86 | 5 | 0.054945 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.003226 | 0 | 3 | 0.346747 | 0 | 88 | 10 | 0.102041 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 3 | 0.266786 | 1 | 56 | 4 | 0.081967 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 3 | 0.245501 | 0 | 84 | 4 | 0.045455 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.007048 | 0 | 2 | 0.037209 | 0 | 66 | 4 | 0.057143 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 2 | 0.111125 | 0 | 86 | 3 | 0.033708 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.00268 | 0 | 2 | 0.004931 | 0 | 94 | 2 | 0.020833 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 4 | 0.01625 | 0 | 2 | 0.071157 | 0 | 98 | 4 | 0.039216 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.010797 | 0 | 3 | 0.267574 | 0 | 46 | 6 | 0.115385 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.014757 | 0 | 0 | 0 | 0 | 48 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 64 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.020485 | 0 | 2 | 0.02896 | 0 | 72 | 4 | 0.052632 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.001787 | 0 | 0 | 0 | 0 | 100 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 2 | 0.039621 | 0 | 84 | 2 | 0.023256 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.003358 | 0 | 0 | 0 | 0 | 58 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.014091 | 0 | 4 | 0.323802 | 0 | 74 | 6 | 0.075 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 62 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 2 | 0.044686 | 0 | 74 | 4 | 0.051282 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.007059 | 0 | 2 | 0.006393 | 0 | 80 | 2 | 0.02439 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.011423 | 0 | 2 | 0.034131 | 0 | 88 | 4 | 0.043478 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 2 | 0.049582 | 0 | 84 | 2 | 0.023256 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 4 | 0.025208 | 0 | 0 | 0 | 0 | 78 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 78 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.014098 | 0 | 2 | 0.017713 | 0 | 76 | 2 | 0.025641 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.013559 | 0 | 2 | 0.004493 | 0 | 78 | 2 | 0.025 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.001315 | 0 | 0 | 0 | 0 | 70 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 9.27E-04 | 0 | 0 | 0 | 0 | 76 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 68 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.005596 | 0 | 2 | 0.025171 | 0 | 74 | 2 | 0.026316 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 72 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.006793 | 0 | 5 | 0.426435 | 0 | 62 | 6 | 0.088235 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 2 | 0.01143 | 0 | 76 | 4 | 0.05 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 4 | 0.014454 | 0 | 2 | 0.065829 | 0 | 90 | 8 | 0.081633 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 78 | 0 | 0 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0 | 0 | 2 | 0.039753 | 0 | 76 | 2 | 0.025641 |
| 100 | 30 | 30 | 100 | 2 | 2 | 1 | 2 | 0.001109 | 0 | 2 | 0.024239 | 0 | 80 | 2 | 0.02439 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 16 | 4.031005 | 6 | 2 | 0.067047 | 0 | 160 | 2 | 0.011905 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 18 | 6.989133 | 0 | 2 | 0.160476 | 0 | 168 | 6 | 0.034483 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 18 | 4.388457 | 0 | 8 | 3.10971 | 5 | 156 | 18 | 0.128492 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 14 | 3.605881 | 0 | 2 | 0.134079 | 0 | 182 | 4 | 0.021505 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 26 | 10.23071 | 16 | 9 | 2.810464 | 2 | 160 | 17 | 0.097436 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 8 | 1.067312 | 0 | 3 | 0.177526 | 0 | 130 | 6 | 0.044118 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 12 | 3.408675 | 10 | 18 | 6.503592 | 0 | 142 | 29 | 0.160221 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 32 | 11.03133 | 18 | 6 | 1.539529 | 3 | 174 | 16 | 0.090047 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 14 | 3.104558 | 12 | 5 | 0.77041 | 0 | 150 | 10 | 0.05814 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 26 | 10.97843 | 6 | 4 | 0.621115 | 0 | 176 | 18 | 0.09 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 20 | 6.699294 | 0 | 9 | 2.606937 | 5 | 174 | 14 | 0.098446 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 38 | 8.702293 | 38 | 9 | 2.351472 | 0 | 170 | 14 | 0.063063 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 18 | 5.252914 | 0 | 2 | 0.071556 | 0 | 170 | 4 | 0.022989 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 14 | 3.99572 | 2 | 5 | 0.338581 | 5 | 170 | 12 | 0.089947 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 12 | 2.896025 | 10 | 2 | 0.072452 | 0 | 176 | 4 | 0.021053 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 18 | 3.832679 | 12 | 3 | 0.222751 | 0 | 154 | 8 | 0.045977 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 18 | 5.358662 | 18 | 6 | 0.889335 | 0 | 188 | 10 | 0.046296 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 16 | 2.790342 | 4 | 2 | 0.089785 | 0 | 188 | 6 | 0.030303 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 34 | 16.45043 | 18 | 3 | 0.458645 | 0 | 198 | 14 | 0.06087 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 26 | 9.988809 | 26 | 5 | 0.807427 | 3 | 154 | 8 | 0.057592 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 16 | 4.753131 | 2 | 5 | 0.531717 | 0 | 172 | 14 | 0.074468 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 10 | 1.185199 | 2 | 5 | 0.717951 | 0 | 174 | 12 | 0.06383 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 22 | 7.00165 | 16 | 5 | 0.747572 | 4 | 180 | 9 | 0.062201 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 26 | 4.709939 | 24 | 8 | 2.059875 | 2 | 184 | 15 | 0.075556 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 36 | 9.589089 | 36 | 2 | 0.075267 | 0 | 138 | 6 | 0.033333 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 20 | 6.551239 | 0 | 5 | 0.287886 | 5 | 174 | 4 | 0.04918 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 30 | 6.666617 | 4 | 3 | 0.443278 | 0 | 148 | 12 | 0.073171 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 18 | 8.272169 | 12 | 3 | 0.334968 | 0 | 182 | 6 | 0.03 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 18 | 8.724642 | 14 | 3 | 0.615616 | 1 | 176 | 6 | 0.035533 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 14 | 3.395634 | 0 | 3 | 0.091494 | 0 | 170 | 8 | 0.044944 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 22 | 4.410094 | 16 | 4 | 0.18663 | 3 | 168 | 4 | 0.036649 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 24 | 9.914871 | 10 | 2 | 0.120071 | 0 | 176 | 8 | 0.041237 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 32 | 14.97768 | 24 | 2 | 0.036368 | 0 | 184 | 6 | 0.028037 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 14 | 5.334966 | 4 | 2 | 0.017993 | 0 | 174 | 2 | 0.011111 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 28 | 10.44039 | 12 | 6 | 1.304399 | 3 | 168 | 16 | 0.095477 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 30 | 12.56167 | 10 | 8 | 2.070309 | 8 | 180 | 17 | 0.116279 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 20 | 4.14728 | 0 | 5 | 0.386911 | 0 | 160 | 12 | 0.069767 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 42 | 11.84934 | 28 | 6 | 0.512574 | 5 | 158 | 8 | 0.065327 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 26 | 13.29215 | 14 | 2 | 0.140771 | 1 | 176 | 8 | 0.045226 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 18 | 5.316957 | 2 | 2 | 0.08363 | 1 | 150 | 2 | 0.019355 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 28 | 13.01007 | 16 | 11 | 4.397042 | 7 | 184 | 12 | 0.086758 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 36 | 16.68436 | 28 | 3 | 0.158364 | 0 | 164 | 8 | 0.04 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 26 | 12.64137 | 20 | 3 | 0.229743 | 0 | 178 | 10 | 0.048077 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 32 | 9.970026 | 8 | 8 | 0.753571 | 8 | 194 | 7 | 0.069124 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 32 | 12.48971 | 12 | 2 | 0.078217 | 0 | 172 | 6 | 0.031579 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 24 | 10.39274 | 18 | 4 | 1.109841 | 0 | 186 | 14 | 0.06422 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 22 | 4.881129 | 20 | 11 | 3.485013 | 1 | 164 | 22 | 0.111111 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 36 | 15.2466 | 36 | 2 | 0.035535 | 0 | 140 | 5 | 0.027624 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 22 | 8.072439 | 16 | 3 | 0.176855 | 0 | 162 | 8 | 0.043011 |
| 40 | 40 | 40 | 140 | 3 | 1 | 1 | 30 | 7.888433 | 14 | 2 | 0.009124 | 0 | 170 | 2 | 0.010753 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.660311 | 0 | 97 | 53.18566 | 97 | 122 | 38 | 0.525292 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.447949 | 0 | 121 | 47.16495 | 121 | 130 | 40 | 0.553265 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.727171 | 0 | 96 | 45.19139 | 96 | 134 | 29 | 0.482625 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.547558 | 0 | 74 | 31.69143 | 73 | 136 | 46 | 0.466667 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.507913 | 0 | 102 | 51.33521 | 102 | 132 | 45 | 0.526882 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 4 | 0.302086 | 0 | 82 | 47.91609 | 78 | 132 | 41 | 0.474104 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.341437 | 0 | 78 | 33.37928 | 78 | 136 | 41 | 0.466667 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 4 | 0.502552 | 0 | 79 | 41.35548 | 79 | 122 | 42 | 0.497942 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 10 | 0.664842 | 0 | 86 | 50.25162 | 84 | 140 | 41 | 0.471698 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.694966 | 2 | 90 | 45.66809 | 90 | 124 | 37 | 0.501976 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 10 | 1.136648 | 0 | 109 | 40.58072 | 109 | 166 | 34 | 0.462783 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.709181 | 0 | 58 | 28.21317 | 54 | 116 | 53 | 0.479821 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 2 | 0.06776 | 0 | 88 | 37.24341 | 84 | 136 | 37 | 0.470817 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 4 | 0.384393 | 0 | 125 | 54.96887 | 125 | 148 | 40 | 0.527157 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 4 | 0.39649 | 0 | 90 | 44.76255 | 90 | 136 | 31 | 0.470817 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 10 | 1.087359 | 0 | 92 | 42.14914 | 91 | 136 | 42 | 0.494424 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 4 | 0.184773 | 0 | 71 | 41.97384 | 69 | 130 | 46 | 0.469388 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.929063 | 0 | 102 | 56.28978 | 100 | 148 | 37 | 0.480702 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.510235 | 0 | 98 | 56.83694 | 96 | 138 | 37 | 0.490775 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.558235 | 4 | 85 | 51.32532 | 84 | 140 | 45 | 0.472527 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 10 | 1.439525 | 8 | 72 | 37.16637 | 69 | 130 | 46 | 0.454545 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 12 | 1.09 | 0 | 124 | 66.09137 | 119 | 160 | 32 | 0.485531 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 4 | 0.152251 | 0 | 27 | 6.304243 | 26 | 118 | 33 | 0.333333 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 10 | 0.79354 | 0 | 73 | 32.78412 | 72 | 144 | 31 | 0.417004 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.816137 | 0 | 82 | 43.64932 | 82 | 156 | 49 | 0.456446 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 4 | 0.539637 | 0 | 68 | 37.67007 | 68 | 144 | 49 | 0.448276 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.476597 | 0 | 109 | 54.5829 | 109 | 136 | 40 | 0.522807 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 4 | 0.478815 | 0 | 64 | 35.4018 | 64 | 124 | 41 | 0.458515 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.40634 | 0 | 48 | 30.6148 | 45 | 152 | 42 | 0.364017 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.644572 | 0 | 64 | 33.60567 | 64 | 126 | 49 | 0.472803 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.270531 | 0 | 69 | 24.17366 | 69 | 114 | 46 | 0.502183 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 10 | 0.945041 | 4 | 111 | 50.00732 | 111 | 118 | 32 | 0.539623 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.651708 | 2 | 84 | 48.34593 | 84 | 150 | 41 | 0.451264 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.442172 | 0 | 93 | 57.49544 | 92 | 126 | 45 | 0.520913 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.462153 | 2 | 73 | 25.58076 | 71 | 124 | 38 | 0.46383 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.496352 | 4 | 81 | 45.52077 | 80 | 134 | 37 | 0.458824 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.615497 | 0 | 82 | 45.14535 | 82 | 126 | 41 | 0.493976 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.155894 | 0 | 107 | 56.24178 | 107 | 120 | 34 | 0.54023 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.855283 | 0 | 67 | 30.35474 | 67 | 148 | 38 | 0.41502 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.645483 | 6 | 103 | 48.58655 | 102 | 122 | 33 | 0.513308 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 4 | 0.263494 | 0 | 72 | 35.13374 | 72 | 146 | 29 | 0.408907 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.705843 | 0 | 126 | 74.20115 | 126 | 134 | 43 | 0.557756 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.405487 | 0 | 82 | 40.43042 | 81 | 124 | 40 | 0.493878 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 4 | 0.21866 | 0 | 81 | 32.85735 | 81 | 128 | 36 | 0.477551 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.750175 | 0 | 126 | 52.20526 | 124 | 126 | 47 | 0.575758 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 4 | 0.202077 | 0 | 124 | 73.79563 | 121 | 130 | 34 | 0.54386 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.577858 | 0 | 62 | 31.846 | 62 | 126 | 35 | 0.434978 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 8 | 0.53729 | 0 | 106 | 58.68341 | 106 | 134 | 35 | 0.512727 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.820626 | 0 | 97 | 50.28039 | 97 | 136 | 52 | 0.522807 |
| 30 | 30 | 30 | 100 | 1 | 1 | 1 | 6 | 0.353433 | 0 | 109 | 60.09904 | 108 | 134 | 35 | 0.516245 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.075133 | 0 | 86 | 36.99624 | 83 | 214 | 93 | 0.451282 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 4.02E-04 | 0 | 109 | 54.32611 | 108 | 170 | 108 | 0.559585 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 10 | 0.154844 | 0 | 94 | 49.67153 | 94 | 202 | 112 | 0.504902 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.007841 | 2 | 56 | 21.84789 | 56 | 184 | 112 | 0.474576 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.025854 | 0 | 100 | 45.39577 | 97 | 210 | 81 | 0.458763 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.060011 | 0 | 96 | 50.2376 | 88 | 208 | 82 | 0.449735 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.210894 | 0 | 128 | 71.11774 | 120 | 180 | 86 | 0.533679 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 6 | 0.164659 | 0 | 86 | 48.07813 | 84 | 200 | 98 | 0.47644 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.088017 | 0 | 92 | 51.73524 | 88 | 212 | 100 | 0.47 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 8 | 0.139987 | 0 | 96 | 56.20552 | 96 | 204 | 98 | 0.487437 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.007471 | 0 | 128 | 62.27424 | 126 | 186 | 94 | 0.541872 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.071426 | 0 | 159 | 82.7825 | 159 | 214 | 77 | 0.524444 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.134171 | 0 | 102 | 50.64288 | 100 | 198 | 102 | 0.505 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.048889 | 0 | 61 | 36.23685 | 57 | 190 | 109 | 0.466292 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.041046 | 0 | 73 | 35.58421 | 70 | 190 | 102 | 0.475138 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.019297 | 0 | 117 | 62.8899 | 112 | 212 | 90 | 0.487923 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 8 | 0.234472 | 0 | 119 | 76.68131 | 108 | 206 | 106 | 0.509524 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.007896 | 0 | 84 | 47.01692 | 83 | 198 | 97 | 0.47619 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 6 | 0.200748 | 0 | 103 | 47.35307 | 102 | 200 | 114 | 0.519231 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.082358 | 0 | 115 | 56.88707 | 113 | 182 | 93 | 0.530928 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.008448 | 0 | 130 | 69.75775 | 129 | 202 | 91 | 0.521327 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.001077 | 0 | 154 | 69.33141 | 154 | 206 | 90 | 0.542222 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.023976 | 0 | 101 | 43.24416 | 100 | 178 | 88 | 0.513661 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 8 | 0.360018 | 0 | 112 | 59.10776 | 112 | 204 | 102 | 0.511962 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.052994 | 0 | 91 | 53.84127 | 91 | 208 | 101 | 0.48 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.016476 | 0 | 91 | 52.94356 | 90 | 198 | 94 | 0.481675 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 8 | 0.309476 | 0 | 98 | 41.96469 | 97 | 202 | 109 | 0.504902 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.001913 | 0 | 76 | 47.68157 | 76 | 200 | 106 | 0.47644 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.113833 | 0 | 90 | 38.62992 | 89 | 190 | 91 | 0.486486 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.062801 | 0 | 140 | 66.53203 | 140 | 204 | 104 | 0.544643 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.133849 | 0 | 123 | 46.94789 | 120 | 188 | 82 | 0.517949 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.013795 | 0 | 85 | 41.19286 | 82 | 196 | 108 | 0.492228 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.098308 | 0 | 118 | 63.14607 | 117 | 204 | 101 | 0.516588 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.02237 | 0 | 118 | 60.1377 | 116 | 176 | 98 | 0.548718 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.004768 | 0 | 136 | 74.0504 | 135 | 176 | 97 | 0.568627 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.005504 | 0 | 144 | 64.80649 | 142 | 206 | 80 | 0.518692 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.030881 | 0 | 156 | 85.56268 | 153 | 190 | 101 | 0.572072 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 6 | 0.103548 | 0 | 104 | 62.25941 | 101 | 204 | 103 | 0.5 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 10 | 0.313655 | 0 | 92 | 44.79354 | 90 | 218 | 100 | 0.465686 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.150911 | 0 | 115 | 66.22031 | 113 | 188 | 103 | 0.534653 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.041295 | 0 | 107 | 56.33107 | 106 | 164 | 76 | 0.526012 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.020304 | 0 | 115 | 63.16221 | 113 | 192 | 107 | 0.533981 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.061653 | 0 | 31 | 12.47325 | 31 | 194 | 87 | 0.378205 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.067064 | 0 | 127 | 52.3187 | 127 | 204 | 111 | 0.538462 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.00528 | 0 | 74 | 46.93517 | 69 | 164 | 109 | 0.520468 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.005368 | 0 | 120 | 54.45166 | 120 | 220 | 114 | 0.515419 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.097016 | 0 | 72 | 35.55181 | 68 | 242 | 104 | 0.415459 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 4 | 0.189513 | 0 | 118 | 52.93002 | 118 | 208 | 104 | 0.516279 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 6 | 0.163106 | 0 | 80 | 35.31929 | 77 | 228 | 97 | 0.432836 |
| 20 | 20 | 50 | 80 | 1 | 3 | 2 | 2 | 0.018909 | 0 | 78 | 42.45329 | 70 | 194 | 104 | 0.472826 |