

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації і управління

До захисту допущено:

В.о. завідувача кафедри

(підпис) Олександр ПАВЛОВ
(вл.ім'я, прізвище)

“ ” _____ 2020 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: «Система з підтримки процесу дослідження алгоритмів
розв'язання деякого класу задач теорії розкладів» (комплексна тема)

Загальна частина

Виконали:

студент IV курсу, групи ІС-61

Іваницький Олег Русланович

(прізвище, ім'я, по батькові)

(підпис)

студентка IV курсу, групи ІС-61

Омельченко Інна Олександрівна

(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н., доц. Жданова Олена Григорівна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

**Консультант з
графічної
документації**

доц., к.т.н., доц. Тєлишева Тамара Олексіївна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Рецензент

доц., к.т.н., доц. Рєпнікова Наталія

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Засвідчуємо, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студенти

(підпис)

(підпис)

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл.ім'я, прізвище)

“ ____ ” _____ 2020 р.

**ЗАВДАННЯ
на дипломний проєкт студенту**

Іваницькому Олегу Руслановичу
(прізвище, ім'я, по батькові)

Омельченко Інні Олександрівні
(прізвище, ім'я, по батькові)

1. Тема проєкту «Система з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач теорії розкладів» (комплексна тема).

керівник проєкту Жданова Олена Григорівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “7” травня 2020 р. №1081-с

2. Термін подання студентом проєкту “01” червня 2020 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до

технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. Схема структурна варіантів використання

2. Схема структурна класів

3. Схема структурна послідовності

4. Схема структурна компонентів

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Загальні положення	доц. Жданова О.Г.		
Інформаційне забезпечення	доц. Жданова О.Г.		
Математичне забезпечення	доц. Жданова О.Г.		
Програмне та технічне забезпечення	доц. Жданова О.Г.		
Технологічний розділ	доц. Жданова О.Г.		

7. Дата видачі завдання «13» квітня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення рекомендованої літератури	20.04.2020	
2.	Аналіз існуючих методів розв'язання задачі	25.04.2020	
3.	Постановка та формалізація задачі	30.04.2020	
4.	Розробка інформаційного забезпечення	03.05.2020	
5.	Алгоритмізація задачі	07.05.2020	
6.	Обґрунтування використовуваних технічних засобів	08.05.2020	
7.	Розробка програмного забезпечення	15.05.2020	
8.	Налагодження програми	22.05.2020	
9.	Виконання графічних документів	25.05.2020	
10.	Оформлення пояснювальної записки	31.05.2020	
11.	Подання ДП на попередній захист	15.05.2020	
12.	Подання ДП на основний захист	01.06.2020	
13.	Подання ДП рецензенту	02.06.2020	

Студент

Олег ІВАНИЦЬКИЙ

Студент

Інна ОМЕЛЬЧЕНКО

Керівник

Олена ЖДАНОВА

[illegible]

Пояснювальна записка до дипломного проєкту

на тему: Система з підтримки процесу дослідження алгоритмів
розв'язання деякого класу задач теорії розкладів

Київ – 2020 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з п'яти розділів, містить 8 рисунків, 6 таблиць, 1 додаток, 13 джерел.

Дипломний проект присвячений розробці системи з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач.

Метою розробки є підвищення ефективності процесу дослідження алгоритмів побудови календарного розкладу виконання робіт.

Завдання розробки: створення та реалізація алгоритмів розв'язання задач теорії розкладів, реалізація підсистем візуалізації, класифікації середовища, реалізація генератора вхідних даних, підсистеми автоматичного дослідження алгоритмів та генерації звістності.

У розділі інформаційного забезпечення описано дані та сутності, з якими працює система.

Розділ математичного забезпечення присвячений розробці алгоритмів розв'язання задач теорії розкладів, а також генератора вхідних даних.

Розділ програмного та технічного забезпечення представляє собою опис програмного продукту, створеного з метою автоматизації дослідження алгоритмів теорії розкладів.

У технологічному розділі надано керівництво користувача.

ТЕОРІЯ РОЗКЛАДІВ, РОБОТА, МАШИНА, ДИРЕКТИВНИЙ СТРОК, ВІДХИЛЕННЯ, ЗАПІЗНЕННЯ, ВИПЕРЕДЖЕННЯ, МОМЕНТ ЗАКІНЧЕННЯ, ТРИВАЛІСТЬ, ДІАГРАМА ГАНТА, PYTHON, TKINTER.

					ДП ІС-6110.00.000 ПЗ					
Зм.	Арк.	Прізвище	Підпис	Дата	Система з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач теорії розкладів			Лім.	Лист	Листів
Розробн.		Іваницький О.Р.								
Розробн.		Омельченко І.О.							2	41
Керівн.		Жданова О.Г.						КПІ ім. Ігоря Сікорського Каф. АСОІУ гр. ІС-61		
Н/контр.		Телишева Т.О.								
В.о.зав. каф.		Павлов О.А.								

ABSTRACT

Bachelor's thesis: 8 figures, 4 tables, 1 appendix, 13 references.

Purpose of the work – increasing the efficiency of the process of researching algorithms for constructing a calendar schedule.

Tasks of the work – development, and implementation of algorithms for solving the problems of the scheduling theory, implementation of the subsystem of visualization of schedules, classification of environment, implementation of the generator of input data, implementation of subsystem of automatic analysis of constructed algorithms and reports generation.

In the information support chapter data and entities, that the work is being conducted with, are described.

Chapter of mathematical support is dedicated to the development of algorithms for scheduling and input data generator.

Software chapter describes the program created to automate the research of the algorithms of scheduling theory.

In technological chapter a user guide is provided.

SCHEDULING THEORY, JOB, MACHINE, DUE DATE, TARDINESS, LATENESS, EARLYNESS, FINAL TIME, DURATION, GANTT DIAGRAM, PYTHON.

					ДП ІС-6110.00.000 ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП	6
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	10
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА	10
1.1.1 Опис процесу діяльності	10
1.1.2 Опис функціональної моделі	12
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ	13
1.3 ПОСТАНОВКА ЗАДАЧІ	15
1.3.1 Призначення розробки	15
1.3.2 Цілі та задачі розробки	15
Висновок до розділу	16
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	17
2.1 ВХІДНІ ДАНІ	17
2.2 ВИХІДНІ ДАНІ	20
2.3 СТРУКТУРА МАСИВІВ ІНФОРМАЦІЇ	22
Висновок до розділу	23
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	24
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	25
4.1 ЗАСОБИ РОЗРОБКИ	25
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	26
4.2.1 Загальні вимоги	26
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	27
4.3.1 Діаграма класів	27
4.3.2 Діаграма послідовності	28
4.3.3 Діаграма компонентів	29
4.3.4 Специфікація функцій	30
4.4 ОПИС ЗВІТІВ	30

Висновок до розділу	31
5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	32
5.1 Керівництво користувача	32
5.2 Випробування програмного продукту	33
5.2.1 Мета випробувань	33
5.2.2 Загальні положення.....	33
5.2.3 Результати випробувань	34
Висновок до розділу	34
ЗАГАЛЬНІ ВИСНОВКИ	35
ПЕРЕЛІК ПОСИЛАНЬ	37
ДОДАТОК А	39

ВСТУП

Дипломний проект присвячений розробці системи з підтримки процесу дослідження алгоритмів теорії розкладів, а також алгоритмів для розв'язання деякого класу задач теорії розкладів..

Теорія розкладів – розділ дослідження операцій, який займається проблемою впорядкування. Побудова розкладу є формою прийняття рішення і грає важливу роль у багатьох сферах людського життя. Як писав Джордан Елленберг у своїй книзі «Як ніколи не помилятися. Сила математичного мислення»: «... країни перемагають у війнах не тому, що вони хоробріші противника або більш незалежні, або їм трохи більше благоволить Бог. Як правило, переможцем стає той, у кого збивають на 5% менше літаків, або хто використовує на 5% менше палива, або хто забезпечує піхоті на 5% більш якісне харчування при 95% відсотках витрат» [1]. Саме вміння ефективного планування та розподілу ресурсів і забезпечує ці критичні п'ять відсотків, які можуть змінити усе. На сучасному конкурентному ринку вміння побудови ефективного виробництва має ключову роль у досягненні успіху.

Теорія розкладів широко використовується на практиці з метою збільшення ефективності виробництва вже дуже давно. Проте лише на початку минулого століття за цю науку взялися серйозно завдяки працям Генрі Ганта [2]. А до появи перших публікацій в інженерній та дослідницькій літературі пройшло ще чимало років. В останніх десятиліттях ХХ століття, з розвитком комп'ютерів та програмування, почало з'являтися все більше досліджень та результатів. Зараз, завдяки комп'ютеризації, теорія розкладів активно розвивається і застосовується на практиці у всіх сферах, де присутнє планування, включаючи виробництво, транспорт, будівництво, спорт, інформаційні технології тощо.

Теорія розкладів оперує ресурсами і завданнями, що, відповідно до області застосування, можуть приймати будь-яку форму. Наприклад,

					ДП ІС-6110.00.000 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ресурсами можуть бути верстати на фабриці, ворота в аеропорту, каси в магазині, кадри на будівництві, процесори в комп'ютері. Завданнями в даних ситуаціях будуть, відповідно, процеси виготовлення деталей, зльоти і посадки літаків, процеси обслуговування клієнтів, операції, виконувані будівельниками та виконання комп'ютерних програм. Кожне завдання має свою тривалість, час надходження та директивний строк, тобто час, до якого потрібно це завдання виконати.

Ціллю побудови розкладу є оптимізація однієї чи декількох цільових функцій, що, в залежності від області, також можуть мати багато різних форм. Наприклад, дуже популярною є задача мінімізації кількості робіт, що завершаються після директивних строків, тобто, мінімізувати кількість робіт, що запізнюються. У даній роботі в якості цільової функції було обрано мінімізацію загального відхилення моментів завершення виконання робіт від їхніх директивних строків, тобто часу, який залишився до директивного терміну та часу, на який виконання завдання спізнилося. Така цільова функція було обрана тому, що дуже часто завершення роботи до директивного терміну також є небажаним, наприклад, якщо результати виконання одного завдання потрібні для початку виконання іншого. У такому випадку, при занадто ранньому закінченню виконання, відповідний ресурс буде простоювати, що призведе до непотрібних витрат.

Таким чином, основна задача теорії розкладів полягає в тому, щоб, маючи обмежену кількість робіт, які необхідно виконати, та обмежену кількість машин, які виконують ці роботи, створити такий розклад, який максимізував би цільову функцію, а саме – мінімізував би загальне відхилення моментів закінчення виконання робіт від директивних строків. Складність даної задачі, так само, як багатьох інших задач теорії розкладів, в тому, що, при збільшенні кількості робіт, знайти оптимальний розклад за поліноміальний час – неможливо, тобто задача є NP-складною.

					ДП ІС-6110.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Для розв'язку NP-складних задач використовують точні та наближені методи. Точні методи знаходять оптимальне рішення, проте вони існують не для всіх задач. Наближені методи дозволяють знайти «хороше» рішення, близьке до оптимального. Основною перевагою наближених методів є час їх роботи. Наближені методи не потребують повного перебору варіантів, вони ґрунтуються на певних евристичних і дозволяють знайти задовільне рішення за задовільний час. Саме тому наближені методи набули популярності і активно досліджуються.

Алгоритми, побудовані в цій роботі, є наближеними методами розв'язку задач впорядкування, вони пропонують розклад з оптимальним, або близьким до оптимального, значенням цільової функції.

Отже, метою даної роботи є розробка та реалізація системи, що полегшить та пришвидшить процес дослідження алгоритмів побудови календарного розкладу виконання робіт для деякого класу задач теорії розкладів. Для ефективного аналізу алгоритмів було розроблено генератор задач, що створює вхідні дані, покриваючи при цьому усі можливі часткові випадки. Таке покриття забезпечується підсистемою класифікації середовища. Це дозволяє ефективно перевіряти алгоритми для будь-яких ситуацій.

Для досягнення поставленої мети необхідно виконати наступні задачі:

- розробити алгоритми побудови календарного розкладу виконання робіт для деякого класу задач теорії розкладів;
- реалізувати розроблені алгоритми програмно;
- реалізувати підсистему візуалізації побудованих розкладів;
- реалізувати підсистему класифікації середовища;
- реалізувати генератор вхідних даних;
- реалізувати підсистему автоматичного проведення експериментів;
- дослідити алгоритми на ефективність за допомогою реалізованих підсистем.

Система, побудована у даній роботі, має науковий зміст і призначена для дослідження алгоритмів. Дана система полегшить роботу науковцям, що працюють з методами теорії розкладів, та пришвидшить їхню роботу.

Розроблені алгоритми мають не тільки науковий зміст, але й практичний. Їх можна буде використовувати як для подальшого дослідження теорії розкладів, так і для налагодження календарного планування на виробництві з метою підвищення ефективності його роботи.

З подальшим розвитком комп'ютерів, а в особливості – їхньої швидкодії, алгоритми, реалізовані в даній роботі, можна буде використовувати для побудови оптимального розкладу, що дозволить підприємствам здійснювати планування не просто у прийнятний спосіб, а у ідеальний.

Практичне значення одержаних результатів. Розроблено систему з підтримки процесу дослідження алгоритмів розв'язання задач теорії розкладів, побудовано алгоритми складання календарного розкладу виконання робіт для деякого класу задач теорії розкладів.

Публікації. Результати роботи були опубліковані у 1 тезах доповідей на науково-технічних конференціях [1].

					ДП ІС-6110.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

1.1.1 Опис процесу діяльності

Теорія розкладів широко застосовується в багатьох галузях: у виробництві, менеджменті, військовій справі, ІТ, транспорті, будівництві, освіті тощо. Ресурси та завдання можуть відрізнятися в залежності від сфери використання, проте, методи та алгоритми розв'язання задач однакові або схожі, що ставить саме процес їх розробки та дослідження на перше місце.

У даній роботі в основі стоїть дослідник, завдання якого полягає у створенні та тестуванні алгоритмів для розв'язання задач теорії розкладів. Процес розробки та дослідження алгоритмів не є простим та складається з наступних кроків:

- дослідження умов задачі, її особливостей;
- аналіз існуючих методів розв'язання задачі;
- розробка алгоритму, який дозволить розв'язувати задачу ефективніше;
- математичне підтвердження ефективності алгоритму;
- програмна реалізація алгоритму;
- створення великої кількості експериментів, що покривають усі можливі часткові випадки задачі;
- тестування алгоритму на згенерованих вхідних даних;
- аналіз результатів та формування звітності.

Весь цей процес є досить складним та тривалим. Кожен з вищенаведених пунктів може виконуватись декілька разів перед отриманням прийнятного результату. Основна складність дослідження алгоритмів теорії розкладів полягає у величезній кількості різноманітних ситуацій, які можуть виникнути у сфері застосування. Саме тому критично важливим при тестуванні

					ДП ІС-6110.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

алгоритму є проведення експериментів з максимальним покриттям часткових випадків. Це забезпечить правильну роботу алгоритму у будь-якій ситуації.

У даній роботі в якості алгоритмів для дослідження було обрано клас задач з одним та двома директивними строками. При наявності робіт, які необхідно виконати, та машин, які виконують ці роботи, потрібно побудувати таке впорядкування робіт, щоб сумарне відхилення моментів закінчення виконання робіт від їх директивних строків було мінімальним. Розглядатимуться системи з однією, декількома ідентичними та декількома пропорційними машинами та з одним та двома директивними строками. У якості цільової функції обрано мінімізацію загального відхилення моментів закінчення виконання робіт від відповідних їм директивних строків.

Під загальним відхиленням розуміється сумарне випередження або запізнення моменту закінчення виконання роботи відносно директивного строку. У той час як запізнення є інтуїтивно зрозумілим, проблему випередження варто пояснити. Причиною врахування випередження у відхиленні є наявність таких ситуацій, у яких занадто ранній час закінчення впливає негативно на процес в цілому. Для прикладу можна привести курси в університеті. Якщо є дві дисципліни, перша з яких ґрунтується на другій, то її вивчення є неможливим без попереднього вивчення другої. Таким чином, виконання другої роботи (вивчення другої дисципліни) має завершитись перед початком виконання першої роботи (вивчення першої дисципліни). У протилежному випадку, бажаного результату не буде досягнуто.

Маючи систему, здатну спростити процес дослідження алгоритмів шляхом автоматичної генерації вхідних даних та тестування алгоритмів, дозволить досліднику не тільки швидше та ефективніше проводити аналіз розроблених алгоритмів, але й покращить якість цього аналізу, мінімізувавши помилки, які можуть виникнути при ручному тестуванні через неуважність та брак часу.

					ДП ІС-6110.00.000 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

1.1.2 Опис функціональної моделі

Інформаційна система, створена у роботі, призначена для полегшення та пришвидшення вищевказаного процесу. До її основних можливостей відноситься:

- генерація достатньо великої кількості наборів вхідних даних для дослідження усіх часткових випадків;
- автоматичне проведення експериментів з використанням відповідних алгоритмів;
- дослідження побудованих розкладів та їхня візуалізація;
- аналіз якості роботи алгоритмів (порівняння із нижньою межею);
- виведення звітності.

У межах системи існує лише один актор – дослідник. У таблиці 1.1 представлені дії, які він може виконати за допомогою створеного у роботі програмного продукту. Діаграма варіантів використання наведена у графічних матеріалах.

Таблиця 1.1 – Опис функцій системи

Функція	Опис функції
Додавання алгоритму	Користувач має можливість додавати до програмного продукту свої реалізації алгоритмів шляхом підключення необхідного файлу з кодом
Введення вхідних даних	Користувач має можливість вводити свої дані для тестування алгоритму
Тестування алгоритму	Користувач має можливість запускати виконання алгоритму на свої даних для перевірки коректності роботи
Генерація вхідних даних	Користувач має можливість генерувати будь-яку кількість вхідних даних для дослідження алгоритму

Продовження таблиці 1.1

Функція	Опис функції
Введення параметрів генератора	Користувач має можливість вводити параметри для генерації необхідних йому задач
Автоматичне експериментування	Користувач має можливість проводити автоматичне тестування алгоритмів на згенерованих вхідних даних
Візуалізація побудованих розкладів	Користувач має можливість генерувати графік з побудованим розкладом
Генерація звітності	Користувач має можливість генерувати звітність про роботу алгоритмів

1.2 Огляд наявних аналогів

Теорії розкладів є наукою дуже широкою та популярною. Існують десятки робіт, присвячених дослідженням різноманітних аспектів цієї дисципліни. Проте, у більшості з них автори присвячують увагу певним розділам теорії розкладів, розробці та застосуванню алгоритмів. Так як таких розділів існує величезна кількість, охопити всіх їх неможливо, тому кожен автор розглядає лише один або декілька класів задач. Натомість розробці систем, які б могли полегшити цей процес, уваги особливо ніхто не надає або ж надає, але дуже мало.

Наприклад, у своїй роботі [4] Лазарев та Гафаров пропонують методи проведення експериментів для будь-яких задач теорії розкладів. Однак, опис цієї схеми є дуже коротким та неповним, без наведення конкретних прикладів та результатів.

Автори роботи [5] реалізували генератор вхідних даних, проте досліджують вони задачі іншого класу теорії розкладів та не покривають весь процес дослідження, а лише розглядають створення індивідуальних задач.

Окрім того, існує багато різних інструментів, які широко використовуються на практиці та дозволяють будувати плани виробництва. До таких інструментів можна віднести, наприклад, Production Planning & Detailed Scheduling [6]. Дана система є одним із модулів SAP [7] і вона дозволяє виконувати функції планування та впорядкування.

Подібний функціонал має система VOGBIT [8], яка дозволяє проводити планування виконання робіт у залежності від ситуації на виробництві та формувати календарний розклад.

Обидві вказані системи мають велике практичне значення та широко використовуються на виробництвах, проте, вони абсолютно не підходять для дослідження. Вони не дають можливості досліднику самостійно додавати алгоритми та проводити експерименти над ними. Окрім того, дані програмні рішення є платними, що не є проблемою для підприємств, але сильно заважає науковцям.

Переваги та недоліки кожної із наведених систем, а також порівняння їх із програмним продуктом, що розробляється у даній роботі, наведено у табл. 1.2.

Таблиця 1.2 – Порівняння з наявними аналогами

Функція	Наша система	SAP	VOGBIT
Побудова розкладу для вхідних даних	Так	Так	Так
Генерація наборів вхідних даних	Так	Так	Ні
Побудова діаграми Ганта	Так	Так	Так
Дослідження алгоритмів	Так	Ні	Ні
Порівняння алгоритмів	Так	Ні	Ні
Платність	Ні	Так	Так

Таким чином, потреба у створенні системи з підтримки процесу дослідження алгоритмів теорії розкладів є очевидною. Такий програмний

продукт дозволить науковцям ефективно досліджувати та аналізувати алгоритми для пізнішого їх впровадження у практичні системи.

1.3 Постановка задачі

1.3.1 Призначення розробки

Система з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач теорії розкладів (далі – Система) призначена для підтримки процесу дослідження алгоритмів, які використовуються для складання календарних розкладів виконання робіт на паралельних машинах.

1.3.2 Цілі та задачі розробки

Метою створення Системи є підвищення ефективності процесу дослідження алгоритмів побудови календарного розкладу виконання робіт шляхом реалізації алгоритмів та генератора задач для автоматизації процесу експериментування та дослідження якості роботи.

Для досягнення мети необхідно реалізувати наступні задачі:

а) створити алгоритми складання розкладу виконання робіт:

- 1) для системи з однією машиною та одним директивним строком;
- 2) для системи з однією машиною та двома директивними строками;
- 3) для системи з декількома ідентичними машинами та одним директивним строком;
- 4) для системи з декількома ідентичними машинами та двома директивними строками;
- 5) для системи з декількома пропорційними машинами та одним директивним строком;
- 6) для системи з декількома пропорційними машинами та двома директивними строками;

б) реалізувати підсистему візуалізації розкладів;

					ДП ІС-6110.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

- в) реалізувати підтримку класифікації середовища;
 - 1) за кількістю та типом машин;
 - 2) за кількістю та величиною директивних строків;
- г) реалізувати на основі класифікації середовища генератор індивідуальних задач для експериментів;
- д) реалізувати підсистему автоматичного проведення експериментів на основі індивідуальних задач, створених генератором;
- е) реалізувати підсистему аналізу результатів та генерації звітності.

Висновок до розділу

У даному розділі було описано середовище, де застосовуватиметься розроблена в роботі система, а також процеси, для автоматизації яких система створюється. Було описано наявні праці, які покривають деякі аспекти даної роботи, наведено постановку задачі, що розв'язується, обґрунтовано призначення та цілі розробки.

Таким чином, у даній роботі буде розроблена система, що є сукупністю багатьох підсистем, які в тому числі включають розробку та реалізацію алгоритмів теорії розкладів. Особливістю даної роботи є програмний продукт, який, окрім реалізації алгоритмів розв'язання деякого класу задач теорії розкладів, надає ще й компоненти для ефективного їх дослідження, а саме – підсистеми візуалізації побудованих розкладів, класифікації середовища, генератора вхідних даних, автоматичного експериментування та виведення результатів аналізу.

Існування такого інструменту дозволить не тільки полегшити та пришвидшити процес дослідження алгоритмів побудови розкладів виконання робіт, а й закладе основу для подальшого дослідження цієї теми.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

У межах системи розглядаються дві основні сутності: ресурси та завдання.

Під ресурсами маються на увазі машини, що здатні виконувати завдання. У реальному світі це можуть бути верстати, процесори, сервери тощо. Машини є паралельними та взаємозамінними. Вони відрізняються лише продуктивністю роботи, тобто мають різну швидкість виконання завдань. Машини не потребують часу на налагодження та здатні виконувати завдання одне за одним. Машини можуть простоювати (у певний момент часу не виконувати жодного завдання) та починати роботу в будь-який момент (не обов'язково в момент початку моделювання). У даній роботі будуть розглянуті системи з однією та декількома машинами. Окрім цього, будуть розглянуті задачі з ідентичними машинами, тобто коли швидкість виконання завдань на різних машинах однакова.

Завдання представляють собою роботи, що мають бути виконані. Для аналогії можна привести виробництво деталей, комп'ютерні процеси, виконання запиту клієнта тощо. Кожна робота має свою тривалість виконання, час надходження у систему (у даній роботі час надходження усіх робіт однаковий та дорівнює 0, тобто усі роботи наявні на початку моделювання), пріоритет (у даній роботі пріоритети усіх робіт однакові), а також директивний строк – момент часу, до якого роботу необхідно виконати. Якщо роботи виконується раніше свого директивного строку, говорять, що робота виконана з випередженням, якщо пізніше – з запізненням. Обидва варіанти вважаються проблемними. Час, на який робота випереджає графік, або на який робота запізнюється, додається до сумарного відхилення моментів закінчення виконання робіт від директивних строків. Метою алгоритмів, розглянутих у роботі є мінімізація цього відхилення. Будуть розглянуті два класи робіт: коли

					ДП ІС-6110.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

усі роботи мають спільний директивний строк та коли є два директивних строки (тобто роботи поділяються на два класи, кожен з яких має свій директивний строк).

Таким чином, у роботі будуть розроблені алгоритми для розв'язання задач з наступними постановками:

- одна машина, один директивний строк;
- одна машина, два директивних строки;
- декілька ідентичних машин, один директивний строк;
- декілька ідентичних машин, два директивних строки;
- декілька пропорційних машин, один директивний строк;
- декілька пропорційних машин, два директивних строки.

Задля ефективного дослідження алгоритмів побудови розкладів для даних задач необхідно мати можливість проводити експерименти в різних ситуаціях з різними вхідними даними. Програмний продукт, реалізований у даній роботі, генерує таку кількість вхідних даних, щоб покрити усі можливі часткові випадки.

Проте, система з підтримки процесу дослідження алгоритмів теорії розкладів може працювати у двох режимах. Перший – підготовчий, коли дослідник лише випробовує алгоритм. На даному етапі дослідник вводить дані вручну та перевіряє розклад, побудований його алгоритмом. Основною метою цього режиму є перевірка на те, чи є побудований розклад допустимим. Другий режим роботи системи є основним у даній роботі. Він призначений для експериментування та дослідження. На даному етапі дослідник уже не вводить дані вручну, натомість – генератор вхідних даних сам створює велику кількість даних, з якими будуть працювати алгоритми.

Таким чином, вхідні дані для двох режимів відрізняються.

У першому випадку, на вхід програмному продукту подається файл з усіма необхідними вхідними даними, описаними вище.

У другому режимі, на вхід програмному продукту не подаються ніякі дані, користувач лише обирає клас задач, алгоритми для розв’язання яких він бажає випробувати та, за бажанням, обирає параметри генератора (рис. 2.1). Окрім того, алгоритми для дослідження мають бути реалізовані (запрограмовані) та підключені до основної програми, а також вони мають працювати з форматом, який підтримується програмою – на вхід вони мають мати можливість приймати дані в такому ж форматі (рис. 2.2), як описаний далі текстовий файл.

У сигнатурі методу, в якому реалізовано алгоритм повинно бути:

- num_d – кількість директивних строків;
- jobs – список, що складається зі списків тривалостей робіт з кожної множини;
- d – список значень директивних строків для кожної множини;
- num_machines – кількість машин;
- machines – список значень продуктивностей машин.

Рисунок 2.1 – Приклад екранної форми для введення параметрів генератора вхідних даних

```
def algorithm(num_d, num_jobs, jobs, d, num_machines, machines)
```

Рисунок 2.2 – Приклад сигнатури методу, в якому реалізується алгоритм

2.2 Вихідні дані

Вихідні дані для двох режимів також відрізняються. У першому випадку результатом роботи програми буде побудований розклад, представлений у вигляді списку з кількістю елементів, рівною кількості машин. Кожен з елементів списку, що представляє собою побудований розклад, також є списком. Він включає в себе роботи, що виконуються на певній машині і відповідають певному директивному строку. У разі, коли тестується задача з одним директивним строком, другий елемент списку для кожної машини – порожній список.

Кожна робота представлена комбінацією наступних її характеристик:

- момент початку виконання роботи;
- тривалість виконання роботи.

Окрім того, модуль візуалізації надає можливість представити побудований розклад графічно (рис. 2.3).

У режимі дослідження в якості результату програмний продукт генерує графік з порівнянням досліджуваних алгоритмів (рис. 2.4).

					ДП ІС-6110.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

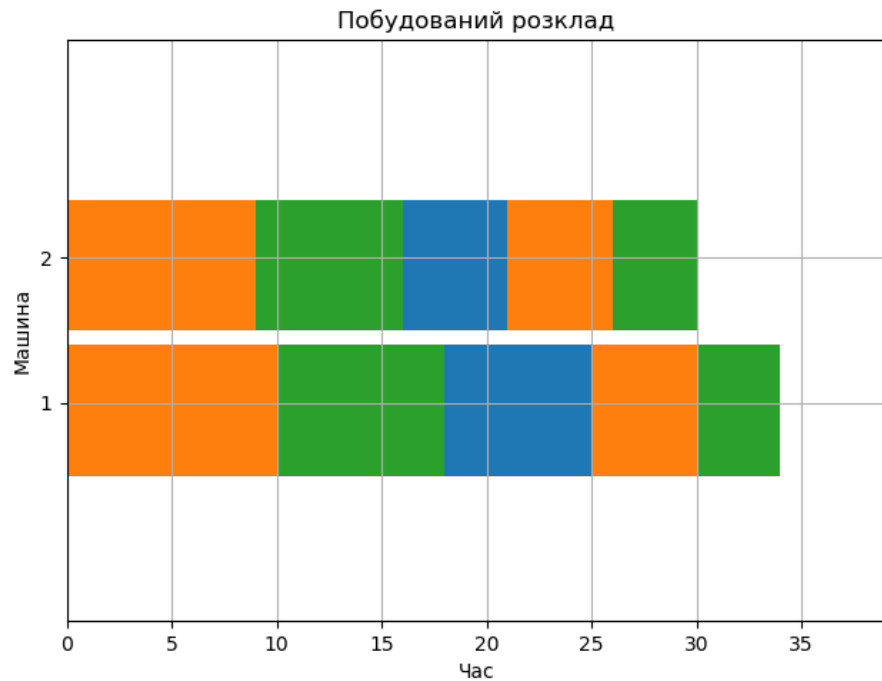


Рисунок 2.3 – Приклад екранної форми з побудованим розкладом

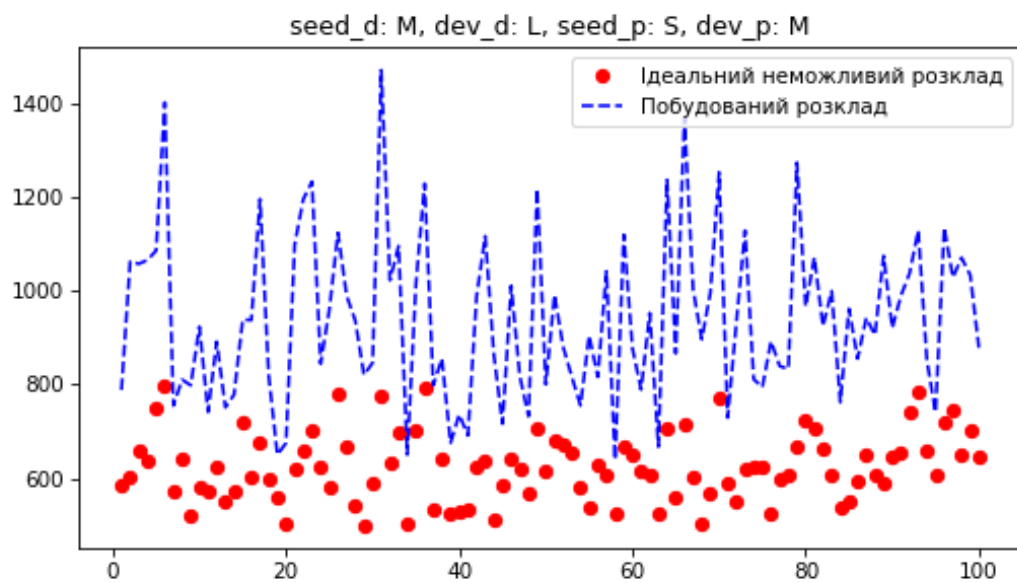


Рисунок 2.4 – Приклад екранної форми з графіком з результатами дослідження алгоритму

2.3 Структура масивів інформації

Як вже зазначалося, програмний продукт приймає вхідні дані лише у першому режимі роботи – при тестуванні алгоритму. У даному випадку для надання значень вхідних даних використовується текстовий файл у формат .txt (рис. 2.5), який містить:

- кількість множин робіт;
- тривалості робіт з першої множини;
- значення першого директивного строку;
- тривалості робіт з другої множини (опціонально, якщо кількість директивних строків дорівнює двом);
- значення другого директивного строку (опціонально);
- кількість машин;
- продуктивності машин (якщо усі машини ідентичні, то їхні продуктивності мають бути однакові).

```

1 2
2 1 4 11 8 9 12 9 10 3
3 20
4 6 6 14 9 1 5 10 4 5 9
5 40
6 3
7 1 1.5 2

```

Рисунок 2.5 – Приклад вхідного текстового файлу для першого режиму роботи програмного продукту

Результат роботи програмного продукту у першому режимі – побудований розклад – також записується у текстовий файл у наступному вигляді (рис. 2.6): перший рядок містить значення директивних строків, потім іде по три рядки для кожної машини. Перший із них містить значення продуктивності машини, другий – роботи з першим директивним строком розділені пробілом, третій – роботи з другим директивним строком (також розділені пробілом). При цьому кожна з робіт записується у вигляді пари

чисел, розділених комою. Першим числом є час початку виконання роботи, другим – тривалість роботи.

1	20 30
2	1
3	6,10 16,4 20,5
4	25,5
5	1
6	8,9 17,3 20,6
7	26,4 30,1
8	1
9	12,8 20,7
10	27,3 30,2

Рисунок 2.6 – Приклад вихідного текстового файлу для першого режиму роботи програмного продукту

Висновок до розділу

У даному розділі було описано розмежування двох режимів роботи програмного продукту, а також описані формати вхідних та вихідних даних, з якими працює та які генерує програмна у кожному з режимів. Було надано приклад вхідних та вихідних даних для одного з режимів та сигнатуру методу, якій повинні відповідати всі реалізації досліджуваних алгоритмів.

Окрім того, варто відзначити, що система, яка розробляється у даній роботі, призначена для підтримки та автоматизації процесу дослідження і вона працює кожного разу з новими, автоматично згенерованими даними, відповідно, вона не потребує даних з попередніх сесій та не зберігає даних для наступних, а результати роботи програмного продукту можуть бути легко відтворені повторним запуском програми. Таким чином, відсутня необхідність використання бази даних у роботі.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

У рамках дипломного проекту розробляється система, призначена для дослідження алгоритмів теорії розкладів. Для даної роботи було обрано задачі з одним та двома директивними строками. Саме над даним класом задач буде проводитись дослідження в межах дипломної роботи.

Задачі поділено між індивідуальними частинами в залежності від кількості директивних строків. У кожній з індивідуальних частин описуються задачі, що розглядаються, а також розроблені алгоритми для їх розв'язання та приклади роботи наведених алгоритмів.

В індивідуальній частині № 1 розглядається клас задач з одним директивним терміном. Наводяться часткові випадки, що можуть виникнути при розв'язанні задачі та відповідні алгоритми для побудови розкладів. Окрім того, в даній частині наводиться алгоритм поділу робіт між машинами (у випадку, коли в системі присутні декілька машин, що можуть виконувати роботи).

В індивідуальній частині № 2 розглядається розширення алгоритмів, розроблених в індивідуальній частині № 1, для випадку задач з двома директивними строками. Окрім того, у цій частині наводиться класифікація середовища, згідно з якою розроблятиметься генератор вхідних даних.

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

При створенні програмного продукту використовувались наступні інструменти: Python, Tkinter, matplotlib, PyCharm, Git. Для моделювання майбутньої системи був використаний Enterprise Architect.

Python – це інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією для загального застосування [9]. Python широко використовується у багатьох сферах, в тому числі і для розробки програмного забезпечення. Низький поріг входу та простота зробили цю мову програмування основною для будь-яких досліджень та проведення дослідів. Саме тому для написання функціоналу було обрано цю мову. Серед її переваг:

- якісна та доступна документація;
- простота написання та чистота коду;
- велика спільнота розробників;
- динамічність;
- потужність;
- наявність великої кількості бібліотек та фреймворків.

Python часто використовується для таких задач як моделювання або експериментування, так як через його простоту можна дуже швидко написати код для будь-якої системи і одразу ж випробувати його. Тільки після успішного випробування програмний продукт переписують з використанням швидших мов програмування, таких як Java чи C++. Так як метою даної роботи є саме спрощення процесу дослідження, то програма не буде переписуватись на інші мови. Python повністю підходить для проведення експериментів та випробування алгоритмів теорії розкладів.

Tkinter – стандартний пакет мови Python, що містить усі необхідні інструменти для розробки графічного інтерфейсу [10]. Tkinter простий у

					ДП ІС-6110.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

використанні, але при цьому дуже потужний. Він надає доступ до усіх необхідних функцій, які використовуються при створенні графічного інтерфейсу користувача.

Matplotlib – графічна бібліотека мови Python [11]. Найпопулярніша бібліотека для побудови графіків та рисунків, яка надає зручний інтерфейс для їх інтеграції в програмні продукти.

PyCharm – популярне середовище розробки, специфіковане під мову програмування Python [12]. Надає весь необхідний інструментарій для розробки програмного забезпечення, а також зручний доступ до системи контролю версій – git [13].

Git використовується для збереження прогресу у розробці програмного продукту з можливістю повернення до його попередніх версій. Це забезпечує ефективну роботу декількох розробників над одним проектом, а також мінімізує ризик втрати прогресу.

Enterprise Architect – інтуїтивно зручний та сучасний засіб моделювання систем, що використовує стандарти UML. Саме з його допомогою були створені всі діаграми, наведені у даній роботі.

4.2 Вимоги до технічного забезпечення

4.2.1 Загальні вимоги

Мінімальні технічні характеристики ЕОМ, необхідні для коректної роботи програмного продукту:

- оперативна пам'ять – 2 Гб або більше;
- жорсткий диск – 20 Мб вільного місця або більше.

Програмні засоби, що використовуються під час проведення випробувань

- операційна система Windows 7 або вище;
- інтерпретатор Python.

4.3 Архітектура програмного забезпечення

4.3.1 Діаграма класів

Програмний продукт розділено на дві складові – графічний інтерфейс користувача та функціональний блок. Графічний інтерфейс призначений для зручного користування та представлений у вигляді вікон. Кожне вікно – окремий крок у дослідженні або тестуванні алгоритмів. Графічний інтерфейс представлений класами, назви та опис яких наведені у таблиці 4.1.

Таблиця 4.1 – Класи графічного інтерфейсу та їхні функції

Клас	Опис
App	Вхідна точка програми, містить у собі посилання на усі інші класи.
MainPage	Головне вікно для вибору подальших дій користувача.
Test	Вікно для вибору файлу з вхідними даними для тестування алгоритму.
ResearchOneDueDate	Вікно для введення параметрів генератора для дослідження алгоритмів розв'язання задач з одним директивним строком.
ResearchTwoDueDates	Вікно для введення параметрів генератора для дослідження алгоритмів розв'язання задач з двома директивними строками.

Функціональний блок відповідає за сам процес тестування та дослідження. Це ядро програми, яке представлено класами, наведеними в таблиці 4.2.

Таблиця 4.2 – Класи функціонального блоку та їхні функції

Клас	Опис
Schedule	Клас для побудови розкладів. Містить у собі усі алгоритми, розроблені у роботі.
Gantt	Клас для візуалізації побудованого розкладу (діаграма Ганта) та виведення його.
Generator	Клас для генерації вхідних даних.

Research	Клас для дослідження алгоритмів та виведення результату.
----------	--

Діаграма класів, що наведена у графічних матеріалах, демонструє відношення класів між собою.

4.3.2 Діаграма послідовності

Як уже зазначалося, програма може працювати у двох режимах – тестування та дослідження. Режим дослідження також розділяється на два незалежних процеси – дослідження задач з одним директивним строком та двома. Таким чином, користувач може в один момент часу виконувати одне з наступного – тестування алгоритму, дослідження задач з одним директивним строком або дослідження задач з двома директивними строками. Кожен з трьох процесів представлений окремою альтернативою на діаграмі діяльності, що наведена у графічних матеріалах. На ній зазначено, як об'єкти програмного продукту взаємодіють між собою та послідовність викликів функцій у кожному з процесів.

Наведемо коротко послідовність виконання дій користувачем у кожному із зазначених процесів. При запуску програми користувач потрапляє на перше вікно (представлене класом MainPage), де він може обрати, що саме він бажає зробити. Після цього послідовність дій у двох режимах відрізняється (при цьому послідовність дій при дослідженні задач з одним та двома директивними строками однакова – різняться лише вікна, що викликаються у кожному з процесів).

У режимі тестування користувач спочатку обирає файл з вхідними даними у вікні графічного інтерфейсу, представленого класом Test, після чого об'єкт класу Gantt отримає шлях до цього файлу, читає його та передає вхідні дані об'єкту класу Schedule, який, відповідно до цих даних, обирає необхідний алгоритм та будує розклад, після чого повертає його у клас Gantt, який будує графік з розкладом, а також записує сам розклад до вихідного файлу.

					ДП ІС-6110.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

У режимі дослідження користувач вводить необхідні йому початкові параметри генератора або обирає варіант «повного дослідження» на вікні інтерфейсу, представленого класом ResearchOneDueDate або ResearchTwoDueDates (в залежності від класу задач, що досліджується). Параметри генератора (при наявності) передаються у об'єкт класу Research, який спочатку за допомогою об'єкта класу Generator генерує велику кількість унікальних наборів вхідних даних, після чого передає їх об'єкту класу Schedule, який буде розклади для зазначених вхідних даних, а також для ідеальних (але неможливих) випадків з цими ж вхідними даними (мається на увазі збільшення директивних строків з метою використання для кожного набору даних алгоритмів, що гарантовано побудують оптимальний розклад). Результуючі значення повертаються в об'єкт класу Research, який буде графік порівняння значень цільових функцій побудованих розкладів та їхніх нижніх границь (для кожного набору вхідних даних отримаємо два значення цільової функції – реальне та ідеальне). Цей графік і є результатом дослідження алгоритмів.

4.3.3 Діаграма компонентів

Структура програмного продукту складається з декількох елементів, серед яких:

- GUI (підсистема графічного інтерфейсу);
- Gantt;
- Research;
- Schedule;
- Generator.

Компоненти Schedule та Generator є надаючими інтерфейсами, тобто надають функціонал елементам Gantt та Research.

Окрім того, у структурі програмного продукту присутні ще два елементи: main (вхідна точка програми) та params (набір констант, які використовуються у програмі).

Діаграма компонентів, на якій зображено відношення елементів програмного продукту, знаходиться у графічних матеріалах.

4.3.4 Специфікація функцій

Усі функції (як графічного інтерфейсу, так і функціонального блоку), окрім двох загальних класів, методи яких наведені у таблицях 4.3 та 4.4, розбиті на дві частини в залежності від автора та представлені у індивідуальних частинах.

Таблиця 4.3 – Опис методів класу App

Метод	Опис методу	Параметри	Опис параметрів
__init__	Конструктор класу, створює вікно	args, kwargs	Параметри запуску (опціонально)

Таблиця 4.4 – Опис методів класу MainPage

Метод	Опис методу	Параметри	Опис параметрів
__init__	Конструктор класу, створює вікно	parent, controller	Батьківський віджет, контролер

4.4 Опис звітів

Як вже зазначалося у розділі 2, програмний продукт може працювати у двох режимах. І результати виконання програми у кожному з них відрізняються. При тестуванні алгоритму результатом роботи є побудований графік, а також його візуалізація – діаграма Ганта. При дослідженні алгоритмів результатом є екранна форма з графіком порівняння значень цільових функцій розкладів, побудованих для згенерованих вхідних даних, та їхніх нижніх границь (при збільшенні директивних строків). Детальний опис звітів наведено у індивідуальних частинах.

Висновок до розділу

У даному розділі було описано засоби розробки програмного продукту: мову програмування, обрану для написання програми (Python), та бібліотеки (Matplotlib) і фреймворки (Tkinter), що застосовуються для створення певних елементів програмного продукту. Було також наведено мінімальні технічні вимоги, необхідні для коректної роботи програмного продукту. Нарешті, було описано архітектуру програмного забезпечення: класи, що застосовуються у програмі, їхні функції, зв'язки один з одним, послідовності виконання викликів методів класів. Було описано, з яких компонентів складається програмний продукт, надано опис результатів виконання програми.

					ДП ІС-6110.00.000 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Керівництво користувача наведено у індивідуальних частинах в залежності від розробленого функціоналу. Спільним є лише головне вікно (рис. 5.1), з якого починається робота з програмою. У залежності від обраної опції на головному вікні викликається наступне, що надає відповідний функціонал. У індивідуальній частині № 1 наведено керівництво користувача у режимі тестування алгоритму, а також при дослідженні алгоритмів для задач з одним директивним строком. У індивідуальній частині № 2 наведено керівництво користувача у режими дослідження для випадку задачі з двома директивними строками.

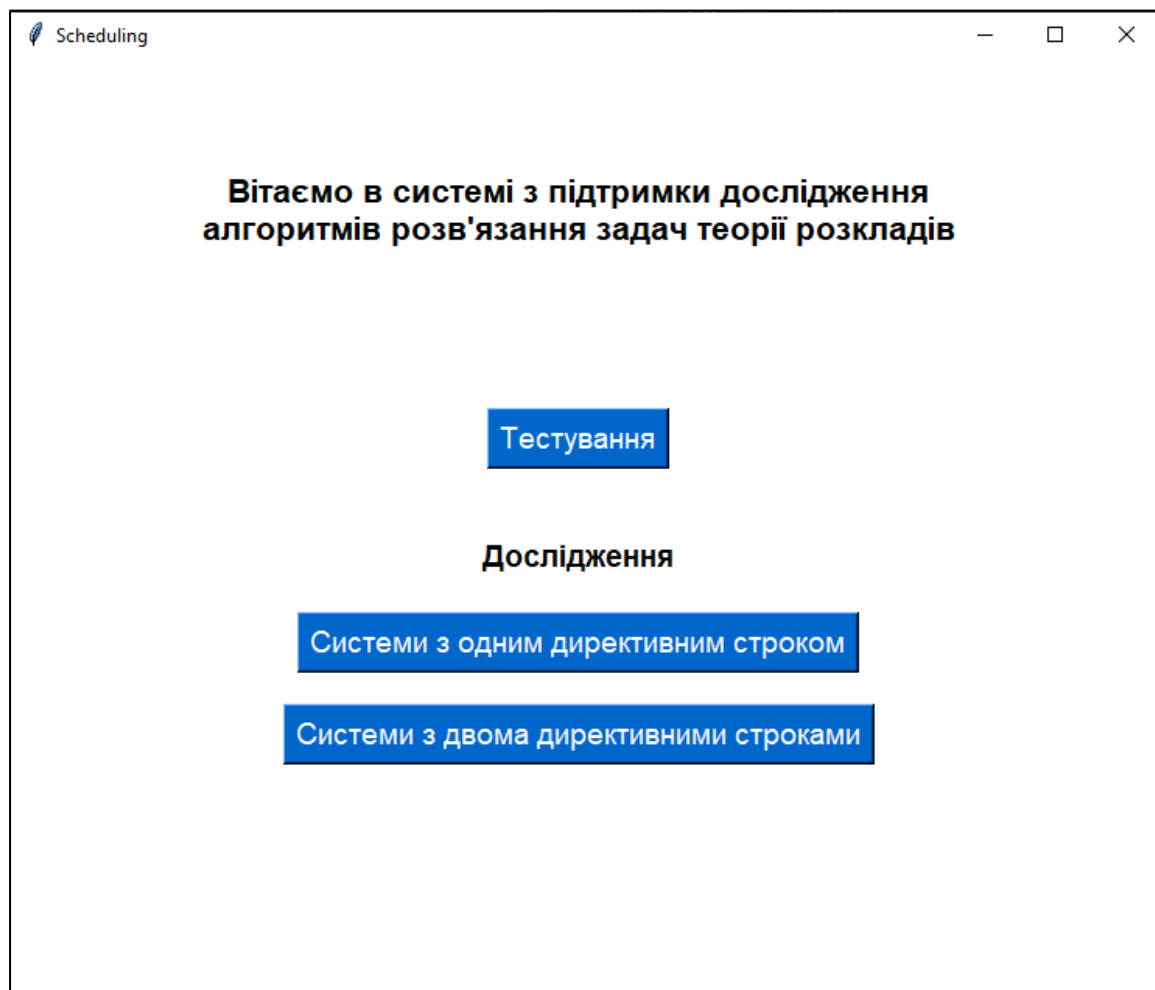


Рисунок 5.1 – Основне вікно програми

5.2 Випробування програмного продукту

5.2.1 Мета випробувань

Метою випробувань є перевірка наявності усіх описаних раніше функцій, перевірка коректності роботи програми та адекватної реакції на виключні ситуації.

5.2.2 Загальні положення

До функцій, які мають бути реалізовані в програмі, належать:

- функція вибору файлу з вхідними даними;
- функція побудови розкладу для заданих вхідних даних;
- функція візуалізації побудованого розкладу у вигляді діаграми Ганта;
- функція введення параметрів генератора вхідних даних;
- функція генерації унікальних наборів вхідних даних;
- функція дослідження алгоритмів.

Для перевірки коректності роботи програмного продукту необхідно виконати усі кроки із керівництва користувача та впевнитись, що усі функції працюють коректно, а виклики – правильні.

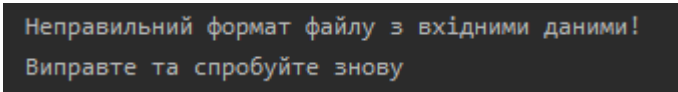
Окрім того, додаткової перевірки може потребувати система побудови розкладу у режимі тестування алгоритмів. Для цього необхідно побудувати розклад вручну, після чого звірити його з розкладом, побудованим програмним шляхом. Дані випробування проведені і описані у індивідуальних частинах.

До виключних ситуацій, що можуть виникнути у ході роботи з програмним продуктом, належить лише випадок, коли обраний файл з вхідними даними не відповідає стандарту, наведеному у розділі 2.

5.2.3 Результати випробувань

Із наведених вище перевірок:

- усі наведені функції реалізовано;
- усі функції виконуються, вікна викликаються у правильному порядку;
- при неправильному форматі файлу з вхідними даними програма виводить відповідне повідомлення (рис. 5.2).



Неправильний формат файлу з вхідними даними!
Виправте та спробуйте знову

Рисунок 5.2 – Повідомлення про неправильний формат вхідного файлу

Висновок до розділу

У даному розділі мова йшла про випробування програмного продукту, було наведено, що саме має бути перевірено та які є можливі випадки, коли програма не буде працювати, як потрібно. Усі випробування було проведено, підтверджено наявність усіх зазначених функцій, обробку виключної ситуації.

Окрім цього, було надано рисунок з головним вікном програмного продукту, з якого починається робота з програмою. Усі подальші інструкції наведено у індивідуальних частинах.

ЗАГАЛЬНІ ВИСНОВКИ

У даній роботі, присвяченій процесу дослідження алгоритмів розв'язання задач теорії розкладів, було розроблено та реалізовано систему, призначену для підтримки цього процесу. Система має на меті спростити та пришвидшити дослідження та аналіз алгоритмів побудови календарного розкладу виконання робіт на паралельних машинах. У якості цільової функції було обрано мінімізацію сумарного відхилення моментів закінчення виконання робіт від директивних строків. У рамках цієї дипломної роботи було розглянуто задачі з одним та двома директивними строками, проте, за потреби, дана система може бути розширена для дослідження алгоритмів розв'язання інших класів задач теорії розкладів: з іншими умовами, обмеженнями та цільовими функціями.

Для створення даної системи було розроблено та реалізовано алгоритми розв'язання задач з одним та двома директивними строками, генератор вхідних даних, підсистеми візуалізації побудованих розкладів та автоматичного експериментування. Окрім того, було створено графічний інтерфейс користувача для полегшення роботи з програмою.

Усі функціональні вимоги, поставлені до програмного продукту, були виконані, програма працює коректно та послідовно, вміє адекватно реагувати на виключні ситуації, що можуть виникнути у процесі роботи з нею. Загалом система виконує свою функцію, має зручний інтерфейс, проста у використанні.

Окрім того, алгоритми, розроблені у процесі виконання роботи, є евристичними методами розв'язання деяких задач теорії розкладів. І хоча вони не завжди будують оптимальний розклад, вони, як було показано при дослідженні алгоритмів, дають непогане близьке до оптимального значення цільової функції за прийнятний час. Саме це і мають виконувати наближені методи розв'язання будь-яких задач і з цією функцією алгоритми, створені у

					ДП ІС-6110.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

даній роботі, чудово справляються. Дані алгоритми надалі можна використовувати при розв'язанні вже зазначених задач теорії розкладів, а також для подальшого їх дослідження та покращення.

У ході дипломної роботи було розглянуто покроково процес створення програмного продукту.

Розділ із загальними положеннями наводить загальний процес дослідження алгоритмів, причини для його автоматизації, кому це може стати корисним, порівняльний аналіз зі схожими наявними програмними продуктами.

Розділ інформаційного забезпечення описує дані, з якими працює програма, їхній вигляд та формат.

Розділ математичного забезпечення присвячений розробці алгоритмів побудови календарного розкладу виконання робіт, а також допоміжних алгоритмів (наприклад, для поділу робіт між машинами). Окрім цього, в даному розділі наведено класифікацію середовища, на основі якої будується генератор вхідних даних.

Розділ програмного та технічного забезпечення описує архітектуру програмного продукту, класи та функції, що використовуються в ньому, послідовність виконання методів та викликів, звіти, що представляють собою результати виконання програми.

Нарешті, у технологічному розділі, наведено керівництво користувача, опис випробувань, проведених над продуктом, їхні результати.

У цілому, варто відзначити, що продукт виконує свої функції підтримки процесу дослідження алгоритмів теорії розкладів, а тому може знайти широке застосування на практиці, що і було метою його створення.

ПЕРЕЛІК ПОСИЛАНЬ

1. Джордан Елленберг. Як ніколи не помилятися. Сила математичного мислення / пер. з англ. Андрій Іщенко. – К.: Наш формат, 2017. – 408 с.
2. Gantt H. L. Work, wages, and profits; their influence on the cost of living / Henry Laurence Gantt. – New York: The Engineering magazine, 1910. – 194 р.
3. Іваницький О.Р., Омельченко І.О., Жданова О.Г. Задача складання розкладу виконання робіт ідентичними паралельними машинами з мінімізацією сумарного відхилення моментів закінчення виконання робіт від директивних строків. Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління – ІСТУ-2020». Секція кафедри автоматизованих систем обробки інформації і управління. Матеріали конференції. – Київ. – 2020. 24, 30 квітня 2020р. – с. 43-47.
4. Лазарев А. Теория расписаний: задачи суммарного запаздывания для одного прибора / А. Лазарев, Е. Гафаров., 2011. – 96 с.
5. Коростильов Є. Ю., Федорко Л. А. Комплекс задач інформаційної підтримки процесу дослідження задач теорії розкладів. – Київ: Національний Технічний Університет України «Київський Політехнічний Інститут імені Ігоря Сікорського», 2018. – 62 с.
6. Sven Denecken, Production Planning and Detailed Scheduling - PP / DS [Електронний ресурс] // Режим доступу: <https://blogs.sap.com/2017/02/03/s4hana-1610-use-case-series-9b-production-planning-and-detailed-scheduling-ppds-tech-view>
7. About SAP SE [Електронний ресурс] // Режим доступу: <https://www.sap.com/corporate/en.html>
8. VOGBIT. Система управления производством. Документация [Електронний ресурс] // Режим доступу: <http://www.vogbit.ru>

9. Python [Електронний ресурс] – Режим доступу до ресурсу: <https://www.python.org/>.
10. Tkinter [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/library/tkinter.html>
11. Matplotlib [Електронний ресурс] – Режим доступу до ресурсу: <https://matplotlib.org/>.
12. Pycharm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jetbrains.com/pycharm/>.
13. Git [Електронний ресурс] – Режим доступу до ресурсу: <https://git-scm.com/>.

ДОДАТОК А

Тексти програмного коду

*Система з підтримки процесу дослідження алгоритмів розв'язання
деякого класу задач теорії розкладів*

(Найменування програми (документа))

CD-RW

(Вид носія даних)

41 арк. 233 КБ

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2020 року

					ДП ІС-6110.00.000 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

```

from params import *
from MainPage import MainPage
from Test import Test
from ResearchOneDueDate import ResearchOneDueDate
from ResearchTwoDueDates import ResearchTwoDueDates

class App(tkinter.Tk):
    def __init__(self, *args, **kwargs):
        tkinter.Tk.__init__(self, *args, **kwargs)

        tkinter.Tk.wm_title(self, 'Scheduling')
        tkinter.Tk.geometry(self, '740x600')

        container = tkinter.Frame(self)
        container.pack(side="top", fill="both", expand=True)
        container.grid_rowconfigure(0, weight=1)
        container.grid_columnconfigure(0, weight=1)

        self.frames = {}

        self.frames[MainPage] = MainPage(container, self)
        self.frames[MainPage].grid(row=0, column=0, sticky='nsew')

        self.frames[Test] = Test(container, self)
        self.frames[Test].grid(row=0, column=0, sticky='nsew')

        self.frames[ResearchOneDueDate] = ResearchOneDueDate(container, self)
        self.frames[ResearchOneDueDate].grid(row=0, column=0, sticky='nsew')

        self.frames[ResearchTwoDueDates] = ResearchTwoDueDates(container, self)
        self.frames[ResearchTwoDueDates].grid(row=0, column=0, sticky='nsew')

        for _, frame in self.frames.items():
            frame.configure(bg='white')

        self.frames[MainPage].tkraise()

from params import *
from Test import Test
from ResearchOneDueDate import ResearchOneDueDate
from ResearchTwoDueDates import ResearchTwoDueDates

class MainPage(tkinter.Frame):
    def __init__(self, parent, controller):
        tkinter.Frame.__init__(self, parent)

        tkinter.Label(self,
                       text='Вітаємо в системі з підтримки дослідження\палгоритмів
розв\'язання задач теорії розкладів',
                       font=font_header, bg='white').grid(row=0, columnspan=2,
padx=120, pady=70)

        self.grid_rowconfigure(1, minsize=30)

```

```

tkinter.Button(self, text='Тестування', font=font, bg=blue_color, fg='white',
               command=lambda: controller.frames[Test].tkraise()).grid(row=2,
columnspan=2)

self.grid_rowconfigure(3, minsize=30)

tkinter.Label(self, text='Дослідження', font=font_bold,
             bg='white').grid(row=4, columnspan=2, pady=12)

tkinter.Button(self, text='Системи з одним директивним строком', font=font,
              bg=blue_color, fg='white',
              command=lambda:
controller.frames[ResearchOneDueDate].tkraise()).grid(row=5, columnspan=2, pady=10)
tkinter.Button(self, text='Системи з двома директивними строками', font=font,
              bg=blue_color, fg='white',
              command=lambda:
controller.frames[ResearchTwoDueDates].tkraise()).grid(row=6, columnspan=2, pady=10)

import tkinter
import matplotlib
matplotlib.use("TkAgg")
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg, NavigationToolbar2Tk
from matplotlib.figure import Figure

font = ('Arial', 14)
font_bold = ('Arial', 14, 'bold')
font_header = ('Arial', 16, 'bold')

blue_color = '#0066cc'
light_grey_color = '#E4E4EB'
dark_grey_color = '#353C4E'

from App import App

def main():
    app = App()
    app.mainloop()

if __name__ == "__main__":
    main()

```

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації і управління

УЗГОДЖЕНО

Керівник проєкту

_____ Олена Жданова
(підпис) (вл. ім'я, прізвище)

“13” квітня 2020 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл. ім'я, прізвище)

“14” квітня 2020 р.

Система з підтримки процесу дослідження алгоритмів розв'язання
деякого класу задач теорії розкладів

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр *ДП 6110.01.000 ТЗ*

на 9 сторінках

Київ – 2020 року

ЗМІСТ

1	ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	3
1.1	Повне найменування системи та її умовне позначення.....	3
1.2	Найменування організації-замовника та організацій-учасників робіт	3
1.3	Перелік документів, на підставі яких створюється система (Завдання на ДП)	3
1.4	Планові терміни початку і закінчення роботи зі створення системи	3
2	ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ.....	4
2.1	Призначення системи	4
2.2	Цілі створення системи	4
3	ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....	6
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	7
4.1	Вимоги до функціональних характеристик.....	7
4.2	Вимоги до надійності	7
4.3	Вимоги до складу і параметрів технічних засобів.....	7
5	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	8
6	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	9
6.1	Види випробувань.....	9

					ДП ІС-6110.01.000 ТЗ			
Зм.	Арк.	Прізвище	Підпис	Дата				
Розробн.		Іваницький О.Р.			Система з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач теорії розкладів	Лім.	Лист	Листів
Розробн.		Омельченко І.О.					2	9
Керівн.		Жданова О.Г.				КПІ ім. Ігоря Сікорського Каф. АСОІУ гр. ІС-61		
Н/контр.		Телишева Т.О.						
В.о.зав. каф.		Павлов О.А.						

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи та її умовне позначення

Система з підтримки процесу дослідження алгоритмів складання календарного розкладу, далі – Система.

1.2 Найменування організації-замовника та організацій-учасників робіт

Замовником системи є кафедра автоматизованих систем обробки інформації і управління факультету інформатики та обчислювальної техніки НТУУ «КПІ імені Ігоря Сікорського». Розробниками системи є Іваницький Олег Русланович та Омельченко Інна Олександрівна.

1.3 Перелік документів, на підставі яких створюється система (Завдання на ДП)

Підставою для розробки Системи є завдання на переддипломну практику.

1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку робіт по створенню Системи – 01.02.2020.
Плановий робіт завершення робіт по створенню Системи – 01.06.2020.

					ДП ІС-6110.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

2.1 Призначення системи

Система призначена для підтримки процесу дослідження алгоритмів, які використовуються для складання календарних розкладів виконання робіт на паралельних машинах.

2.2 Цілі створення системи

Метою створення Системи є підвищення ефективності процесу дослідження алгоритмів побудови календарного розкладу виконання робіт шляхом реалізації алгоритмів та генератора задач для автоматизації процесу експериментування та дослідження якості роботи.

Для досягнення мети необхідно реалізувати наступні задачі:

а) створити алгоритми складання розкладу виконання робіт:

- 1) для середовища з однією машиною та одним директивним строком;
- 2) для середовища з однією машиною та двома директивними строками;
- 3) для середовища з декількома ідентичними машинами та одним директивним строком;
- 4) для середовища з декількома ідентичними машинами та двома директивними строками;
- 5) для середовища з декількома пропорційними машинами та одним директивним строком;
- 6) для середовища з декількома пропорційними машинами та двома директивними строками;

б) реалізувати підсистему візуалізації розкладів;

в) реалізувати підтримку класифікації середовища;

- 1) за кількістю та типом машин;
- 2) за кількістю та величиною директивних строків;

					ДП ІС-6110.01.000 ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

г) реалізувати на основі класифікації середовища генератор індивідуальних задач для експериментів;

д) реалізувати підсистему автоматичного проведення експериментів на основі індивідуальних задач, створених генератором;

е) реалізувати підсистему аналізу результатів та генерацій звітності.

					ДП ІС-6110.01.000 ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Об'єктом автоматизації є процес дослідження алгоритмів розв'язання задач теорії розкладів, а саме – задач календарного планування виконання робіт на паралельних пристроях. Даний процес не є автоматизованим чи уніфікованим, для проведення експериментів досліднику потрібно самому створювати тестові дані, вводити їх у систему та аналізувати результати. При цьому, вхідні дані для експериментів мають бути всеохоплюючими, тобто покривати всі часткові випадки, щоб дослідити якість роботи алгоритмів за будь-яких умов. Це займає велику кількість сил та часу. Зручніше було б мати систему, яка, маючи алгоритми, які потрібно дослідити, здатна була б сама проводити експерименти та генерувати для них вхідні дані, а потім виводити результат про якість досліджуваного алгоритму. Саме тому доцільним є створення програмного продукту, здатного за заданими параметрами сформулювати велику кількість експериментів, провести їх та вивести звітність щодо ефективності та якості алгоритмів.

					ДП ІС-6110.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Система має реалізовувати наступні функції:

- функція побудови розкладу за заданим алгоритмом;
- функція візуалізації побудованого алгоритму;
- функція класифікація середовища;
- функція генерації вхідних даних для експериментів;
- функція проведення експериментів;
- функція перевірки якості роботи алгоритму на великій кількості експериментів;
- функція генерації та виведення звітності.

4.2 Вимоги до надійності

Система має відповідати наступній вимозі до надійності:

- обробка усіх виключних ситуацій.

4.3 Вимоги до складу і параметрів технічних засобів

Мінімальні технічні характеристики ЕОМ, необхідні для коректної роботи програмного продукту:

- оперативна пам'ять – 2 Гб або більше;
- жорсткий диск – 20 Мб вільного місця або більше.

Програмні засоби, що використовуються під час проведення випробувань

- операційна система Windows 7 або вище;
- інтерпретатор Python.

					ДП ІС-6110.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Номер	Стадія	Результат
1	Постановка та формалізація задачі	Технічне завдання
2	Вивчення існуючих методів та алгоритмів	Проаналізовано наявні дослідження
3	Розробка математичної моделі	Розроблена математична модель
4	Розробка програмного продукту	Розроблений програмний продукт
5	Тестування програмного продукту	Підтверджено якість програмного продукту
6	Оформлення документації	Готова документація
7	Здача програмного продукту та документації	Здані програмний продукт та документація

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

6.1 Види випробувань

Задля контролю якості створеного програмного продукту необхідні випробування для наступних функцій:

- функція генерації вхідних даних експериментів;
- функція покриття генератором вхідних даних всіх часткових випадків;
- функція побудови календарного розкладу виконання робіт;
- функція дослідження ефективності роботи алгоритму та порівняння алгоритмів між собою;
- функція формування звітності про ефективність роботи алгоритму;
- функція реагування Системи на виключні ситуації.

					ДП ІС-6110.01.000 ТЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Власник документу:
Попенко Володимир Дмитрович

ID перевірки:
1003783674

Дата перевірки:
04.06.2020 18:31:48 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
04.06.2020 20:56:31 EEST

ID користувача:
77149

Назва документу: Ivanitsky_Omelchenko_bachelor_common

ID файлу: 1003798155 Кількість сторінок: 33 Кількість слів: 6143 Кількість символів: 46135 Розмір файлу: 195.43 KB

9.7% Схожість

Найбільша схожість: 3.91% з джерело https://ela.kpi.ua/bitstream/123456789/30987/1/Zhydkov_bakalavr.pdf

6.58% Схожість з Інтернет джерелами 88 Page 35

9.03% Текстові збіги по Бібліотеці акаунту 398 Page 36

0.83% Цитат

Цитати 1 Page 37

Вилучення переліку посилань вимкнено

0% Вилучень

Вилучений текст відсутній

Підміна символів

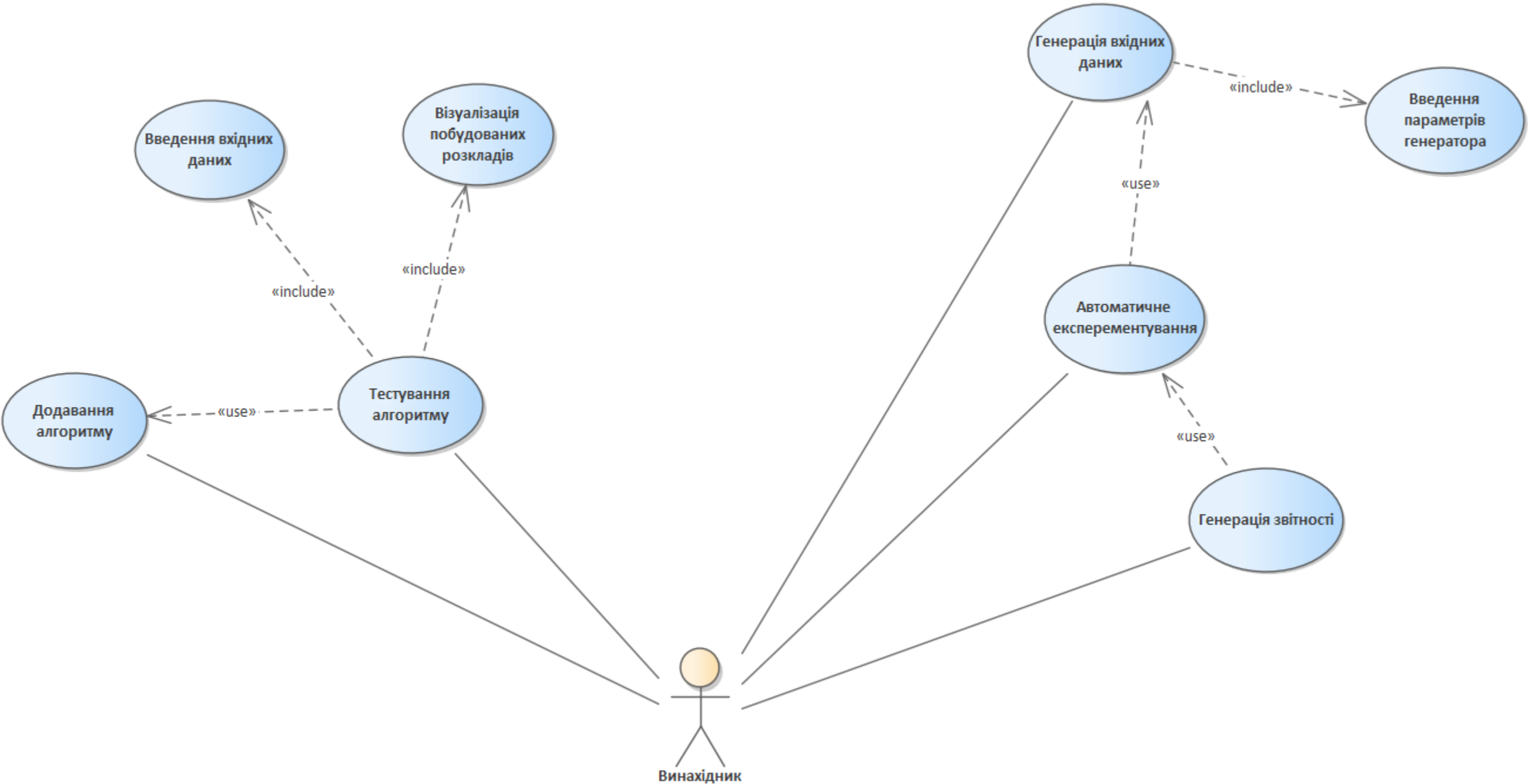
Не знайдено замієених символів

Графічний матеріал до дипломного проєкту

на тему: Система з підтримки процесу дослідження алгоритмів розв'язання
деякого класу задач теорії розкладів

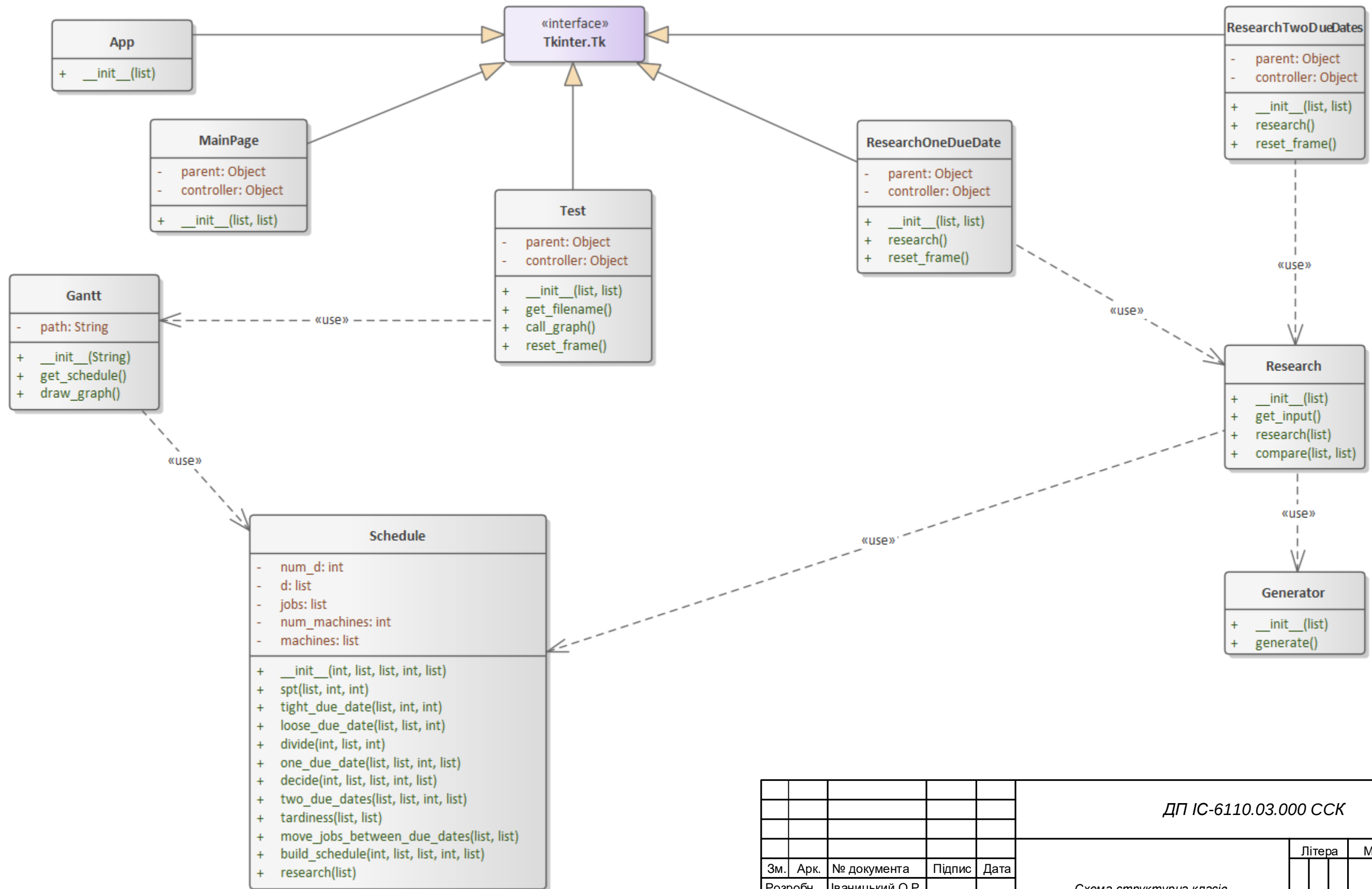
Київ – 2020 року

uc Researcher

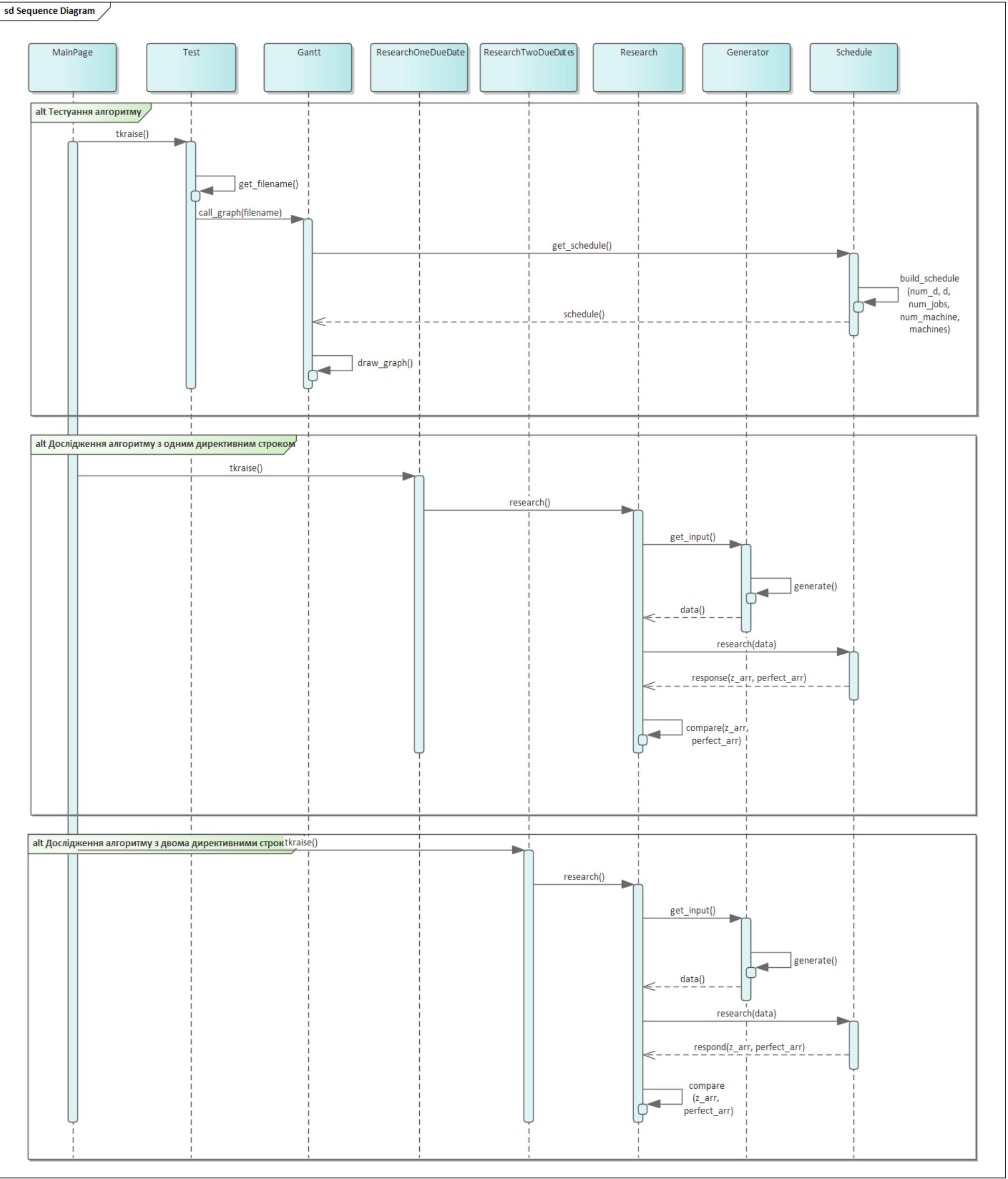


						ДП ІС-6110.02.000 ССВ							
						Схема структурна варіантів використання				Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата									
Розробн.		Іваницький О.Р.											
Розробн.		Омельченко І.О.											
Перевірив		Жданова О.Г.				Система з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач теорії розкладів				Аркуш 1		Аркушів 1	
Т. кон.													
Н. контр.		Телишева Т.О.											
Затвердив		Жданова О.Г.											
						КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-61							

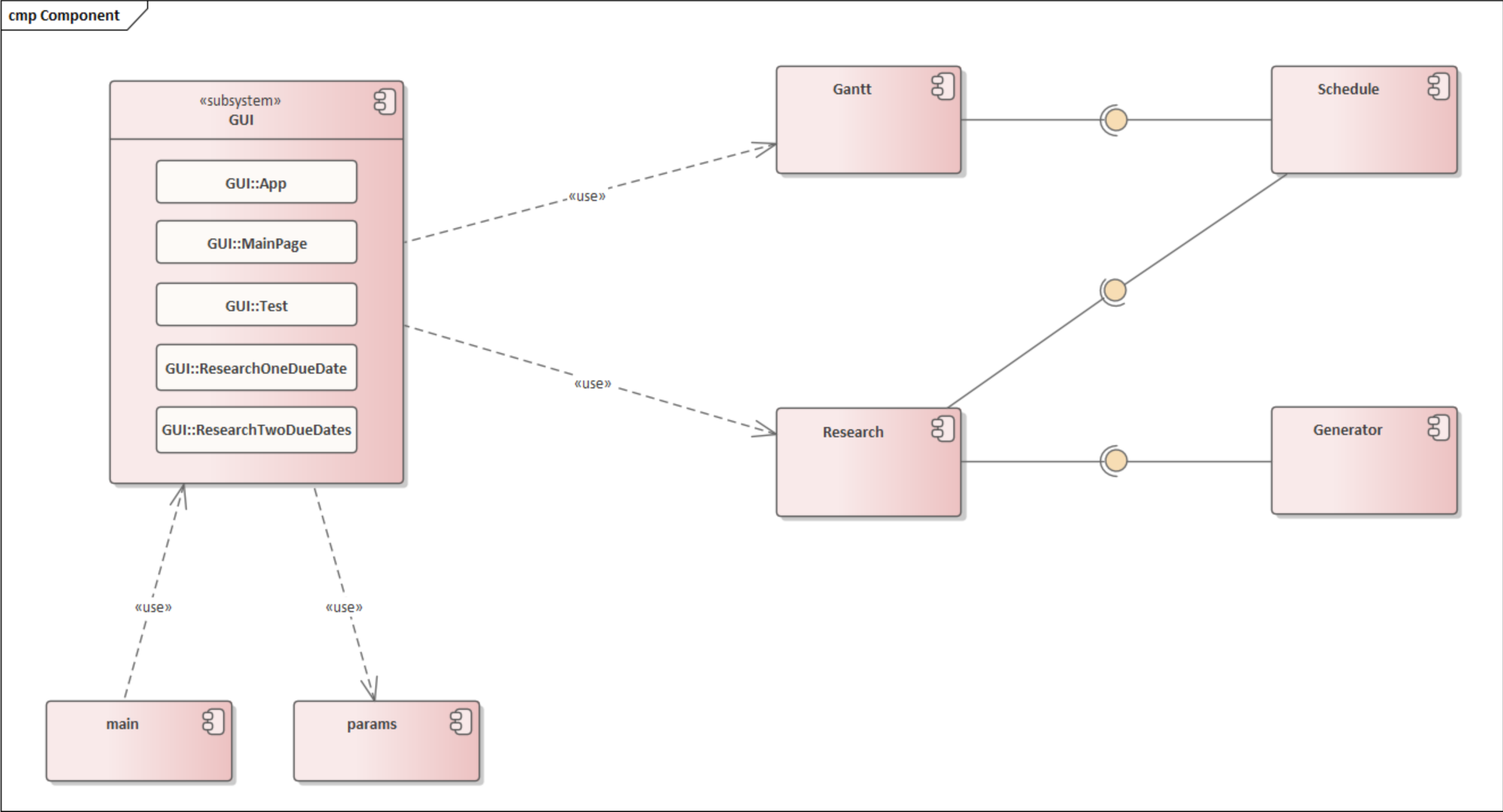
class Class Diagram



					ДП ІС-6110.03.000 ССК					
					Схема структурна класів		Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата						
Розробн.		Іваницький О.Р.								
Розробн.		Омельченко І.О.								
Перевірів		Жданова О.Г.			Система з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач теорії розкладів		Аркуш 1		Аркушів 1	
Т. кон.										
Н. контр.		Телишева Т.О.								
Затвердив		Жданова О.Г.								



					ДП ІС-6110.04.000 ССП			
					Схема структурна послідовності	Лит.	Маса	Масштаб
Зм.	Арк.	№ докум.	Підпис	Дата				
Розробн.		Іваницький О.Р.						
Розробн.		Омельченко І.О.						
Перев.		Жданова О.Г.						
Т. Кон.					Система з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач теорії розкладів	Аркуш 1		Аркушів 1
Н. Контр.		Телишева Т.О.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-61		
Зате.		Жданова О.Г.						



						ДП ІС-6110.05.000 ССК									
						Схема структурна компонентів	Літера			Маса		Масштаб			
Зм.	Арк.	№ документа	Підпис	Дата											
Розробн.		Іваницький О.Р.													
Розробн.		Омельченко І.О.													
Перевірив		Жданова О.Г.													
Т. кон.						Система з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач теорії розкладів	Аркуш 1			Аркушів 1					
Н. контр.		Телишева Т.О.					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-61								
Затвердив		Жданова О.Г.													

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації і управління

До захисту допущено:

В.о. завідувача кафедри

(підпис) Олександр ПАВЛОВ
(вл.ім'я, прізвище)

“ ____ ” _____ 2020 р.

Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: «Система з підтримки процесу дослідження алгоритмів

розв'язання деякого класу задач теорії розкладів» (комплексна тема)

Задачі з одним директивним строком.

Підсистеми візуалізації та аналізу результатів

Індивідуальна частина № 1

Виконала:

студентка IV курсу, групи ІС-61

Омельченко Інна Олександрівна

(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н., доц. Жданова Олена Григорівна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

**Консультант з
графічної
документації**

доц., к.т.н., доц. Тєлишева Тамара Олексіївна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Рецензент

доц., к.т.н., доц. Репнікова Наталія

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ
(підпис) (вл. ім'я, прізвище)

“ ” 2020 р.

**ЗАВДАННЯ
на дипломний проєкт студентці**

Омельченко Інні Олександрівні
(прізвище, ім'я, по батькові)

1. Тема проєкту «Система з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач теорії розкладів» (комплексна тема).

Задачі з одним директивним строком. Підсистеми візуалізації та аналізу результатів (Індивідуальна частина № 1)

керівник проєкту Жданова Олена Григорівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “7” травня 2020 р. №1081-с

2. Термін подання студентом проєкту “01” червня 2020 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань

програмного продукту

5. Перелік графічного матеріалу

1. Схема структурна діяльності

2. Креслення вигляду звітних форм

3. Рішення з математичного забезпечення

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Загальні положення	доц. Жданова О.Г.		
Інформаційне забезпечення	доц. Жданова О.Г.		
Математичне забезпечення	доц. Жданова О.Г.		
Програмне та технічне забезпечення	доц. Жданова О.Г.		
Технологічний розділ	доц. Жданова О.Г.		

7. Дата видачі завдання «13» квітня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення рекомендованої літератури	20.04.2020	
2.	Аналіз існуючих методів розв'язання задачі	25.04.2020	
3.	Постановка та формалізація задачі	30.04.2020	
4.	Розробка інформаційного забезпечення	03.05.2020	
5.	Алгоритмізація задачі	07.05.2020	
6.	Обґрунтування використовуваних технічних засобів	08.05.2020	
7.	Розробка програмного забезпечення	15.05.2020	
8.	Налагодження програми	22.05.2020	
9.	Виконання графічних документів	25.05.2020	
10.	Оформлення пояснювальної записки	31.05.2020	
11.	Подання ДП на попередній захист	15.05.2020	
12.	Подання ДП на основний захист	01.06.2020	
13.	Подання ДП рецензенту	02.06.2020	

Студентка

Інна ОМЕЛЬЧЕНКО

Керівник

Олена ЖДАНОВА

Пояснювальна записка до дипломного проєкту

на тему: Система з підтримки процесу дослідження алгоритмів
розв'язання деякого класу задач теорії розкладів. Задачі з одним
директивним строком. Підсистеми візуалізації та аналізу результатів

Київ – 2020 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка індивідуальної частини дипломного проекту складається з трьох розділів, містить 23 рисунки, 8 таблиць, 1 додаток, 4 джерела.

Дана індивідуальна частина дипломного проекту присвячена реалізації алгоритмів розв'язання задач теорії розкладів з одним директивним строком, а також їх розширенню на задачі з декількома паралельними машинами. Окрім цього, в даній частині розглянуто створення підсистем візуалізації та аналізу результатів дослідження.

Розділ програмного та технічного забезпечення надає опис функцій, реалізованих в рамках даної індивідуальної частини.

У технологічному розділі надано інструкції з використання програмного продукту.

ТЕОРІЯ РОЗКЛАДІВ, РОБОТА, МАШИНА, ДИРЕКТИВНИЙ СТРОК, ВІДХИЛЕННЯ, ТРИВАЛІСТЬ, ДІАГРАМА ГАНТА.

					ДП ІС-6119.00.001 ПЗ			
Зм.	Арк.	Прізвище	Підпис	Дата	Система з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач теорії розкладів. Задачі з одним директивним строком. Підсистеми візуалізації та аналізу результатів	Лім.	Лист	Листів
Розробн.		Омельченко І.О.					2	40
Керівн.		Жданова О.Г.				КПІ ім. Ігоря Сікорського Каф. АСОІУ гр. ІС-61		
Н/контр.		Телишева Т.О.						
В.о.зав. каф.		Павлов О.А.						

ABSTRACT

Bachelor's thesis: 23 figures, 8 tables, 1 appendix, 4 references.

This individual part of diploma is dedicated to the development and programming implementation of algorithms for solving scheduling tasks with one due date. It also gives an extension for these algorithms which makes it possible to use them for several parallel machines. Apart from that, this section of diploma gives explanations to the process of development of visualization and results analysis subsystems.

Software chapter describes all the methods, programmed in this individual part.

In technological chapter instructions for correct program use are provided.

SCHEDULING THEORY, JOB, MACHINE, DUE DATE, TARDINESS,
DURATION, GANTT DIAGRAM.

					ДП ІС-6119.00.001 ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП	5
1 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	7
1.1 Змістовна постановка задачі	7
1.2 Математична постановка задачі	7
1.3 Обґрунтування методу розв'язання	8
1.4 Опис методу розв'язання	8
Висновок до розділу	16
2 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	18
2.1 Архітектура програмного забезпечення	18
2.1.1 Специфікація функцій	18
2.2 Опис звітів	21
Висновок до розділу	23
3 ТЕХНОЛОГІЧНИЙ РОЗДІЛ	24
3.1 Керівництво користувача	24
3.2 Випробування програмного продукту	29
3.2.1 Мета випробувань	29
3.2.2 Загальні положення	29
3.2.3 Результати випробувань	29
Висновок до розділу	32
ЗАГАЛЬНІ ВИСНОВКИ	33
ПЕРЕЛІК ПОСИЛАНЬ	34
ДОДАТОК А	35

ВСТУП

Система з підтримки процесу дослідження алгоритмів розв'язання задач теорії розкладів – це комплексний продукт, який складається з багатьох підсистем, що функціонують разом та доповнюють одне одного. Однією з підсистем є саме розв'язання задач теорії розкладів. У даній роботі обрано клас задач з одним та двома директивними строками. Для забезпечення коректного функціонування системи необхідно розглянути задачі цього класу та розробити алгоритми їх розв'язання. У даній індивідуальній частині мова піде про задачі з одним директивним строком, тобто коли усі роботи системи мають однакове значення директивного строку. Ця задача є простішою, ніж задача з двома директивними строками, проте, алгоритми, розроблені та реалізовані в даній індивідуальній частині, стануть основою для алгоритмів розв'язання задач з двома директивними строками, так як умова з двома множинами робіт є розширенням задачі, розглянутої в даній частині.

Задачі з одним директивним строком є досить поширеними в теорії розкладів: їм присвячені праці багатьох дослідників, ці задачі часто виникають на практиці. Проте, охопити усі аспекти та нюанси, які виникають при розв'язуванні цієї задачі практично неможливо, так як існує безліч можливих варіацій постановок задачі, а також величезна кількість особливих ситуацій, які виникають через різноманітність вхідних даних, і які вимагають особливого підходу.

Саме тому існує потреба в розробці системи, здатної полегшити процес розгляду усіх ймовірних нюансів, що виникають в цій задачі. Умова з одним директивним строком була вибрана для дослідження не просто так. Ця задача заслуговує особливої уваги через свою поширеність.

Авторів, які досліджували окремі випадки даної задачі можна назвати не так багато. Серед них – Пітер Брукер, який у своїй роботі [1] окрім інших розглядає задачу з мінімізацією суми зваженого випередження та зваженого

					ДП ІС-6119.00.001 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

запізнення від директивного строку на одній машині. Автор пропонує алгоритм оптимального розкладу, проте, при збільшенні кількості машин задача дещо ускладнюється, і запропонований алгоритм вже не гарантує оптимального її вирішення.

У статті [2] розглядається лише один аспект нашої задачі – випадок малого директивного строку. Ця ситуація є частковим випадком і не дає можливості повного дослідження задачі при будь-яких умовах.

Схожу задачу розв’язує Майкл Пінедо у своїй книзі [3]. Автор аналізує задачу мінімізації сумарного відхилення моментів закінчення виконання робіт від директивного строку для системи з однією машиною та одним директивним строком. Ця задача стала основою для розглянутого в даній роботі випадку і саме описаний Майклом алгоритм використовується для побудови розкладу у такій ситуації.

Таким чином, у даній частині буде розглянуто клас задач теорії розкладів з одним директивним строком та цільовою функцією, що мінімізує загальне відхилення моментів закінчення виконання робіт від директивного строку. Будуть розглянуті усі можливі випадки, які можуть виникнути у процесі розв’язання задачі, а також описано спосіб розширення задачі на умову з декількома машинами.

1 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

1.1 Змістовна постановка задачі

Задано множину робіт, яка має свій спільний директивний строк. Задана також певна кількість машин, які працюють паралельно, є взаємозамінними і відрізняються одна від одної продуктивністю виконання робіт. Машини можна впорядкувати за швидкістю виконання роботи. Для кожної машини задано час виконання нею роботи.

Передбачається, що всі роботи надходять одночасно в нульовий момент часу, обслуговування кожної роботи проходить без перерв до завершення. Машини можуть простоювати та починати свою роботу в будь-який момент часу.

Необхідно побудувати такий розклад, який би мінімізував сумарне відхилення моментів закінчення виконання робіт від директивного строку.

Розв'язання цієї задачі розбивається на три менші підзадачі, які є частковими випадками в залежності від кількості та типу машин.

Перша підзадача розглядає систему лише з однією машиною, друга – систему з багатьма ідентичними машинами, третя – систему з багатьма пропорційними машинами, тобто які відрізняються одна від одної продуктивністю.

1.2 Математична постановка задачі

Дано:

- множина робіт: $J = \{j_1, j_2, \dots, j_n\}$ з кількістю робіт n ;
- тривалості виконання робіт: $P = \{p_1, p_2, \dots, p_n\}$;
- m машин ($m \geq 1$);
- продуктивності роботи машин: $h = \{h_1, h_2, \dots, h_m\}$;
- директивний строк: d .

					ДП ІС-6119.00.001 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Цільова функція (мінімізація сумарного відхилення моментів завершення виконання робіт від директивного строку) буде виглядати так:
 $L \rightarrow \min$ (згідно з позначеннями, введеними у роботі [3]).

1.3 Обґрунтування методу розв'язання

Для розв'язання задач, умови яких наведені вище, будуть використані алгоритми, описані в роботах [2] та [3]. Вони будуть розширені та узагальнені для задачі з декількома машинами. У названих роботах доводиться, що дані алгоритми будують оптимальний розклад, а тому їх сміливо можна використовувати для вирішення задач.

Окрім того, буде представлено алгоритм для поділу робіт між машинами. Даний алгоритм не гарантує якнайкращий розподіл, проте окрім повного перебору поки що немає такого алгоритму, який за допустимий час призначав би машинам найкращий набір робіт. Так як при дуже великій кількості вхідних робіт проведення повного перебору є неможливим, то для розв'язання задач з декількома машинами буде використовуватися метод, наведений у даній індивідуальній частині.

1.4 Опис методу розв'язання

Для початку розглянемо задачу з однією машиною. Це досить тривіальна задача, проте, вона ляже в основу усіх подальших розширень та модифікацій.

Отже, у процесі розв'язання задачі з одним директивним строком та однією машиною може виникнути дві ситуації в залежності від розташування директивного строку відносно початку координат (нульового моменту часу, тобто моменту надходження робіт):

- коли директивний строк знаходяться далеко від початку координат (рис. 1.1);

- коли директивний строк знаходиться близько до початку координат (рис. 1.2).

Кажучи «знаходяться близько», мається на увазі, що при заданому директивному строку не можна побудувати довільний розклад, який виконував би довільну кількість робіт до директивного строку. Як буде показано далі, алгоритм, що застосовується для випадку великого директивного строку, не зможе бути використаний при малому директивному строку, а тому різниця між таким поділом розміру директивних строків буде залежати самі від можливості використання даного алгоритму.

На рисунках 1.1-1.2 зображені умовні контури розкладів – їхні схематичні вигляди. Основним, на що варто звернути увагу на даних рисунках, є розміщення контурів відносно початку координат. Саме це визначатиме ситуацію, яка буде розглядатися. Наприклад, на рисунку 1.1 контур знаходиться далеко від початку координат, в той час як на рисунку 1.2 контур стоїть впритул до нього, а більша частина контуру знаходиться справа від директивного строку. Це свідчить про те, що за першим алгоритмом роботи не можуть бути розставлені, а, отже, виникає ситуація два – директивний строк достатньо малий для побудови довільного розкладу. У цьому випадку виконання робіт починається з самого моменту їх надходження в систему.

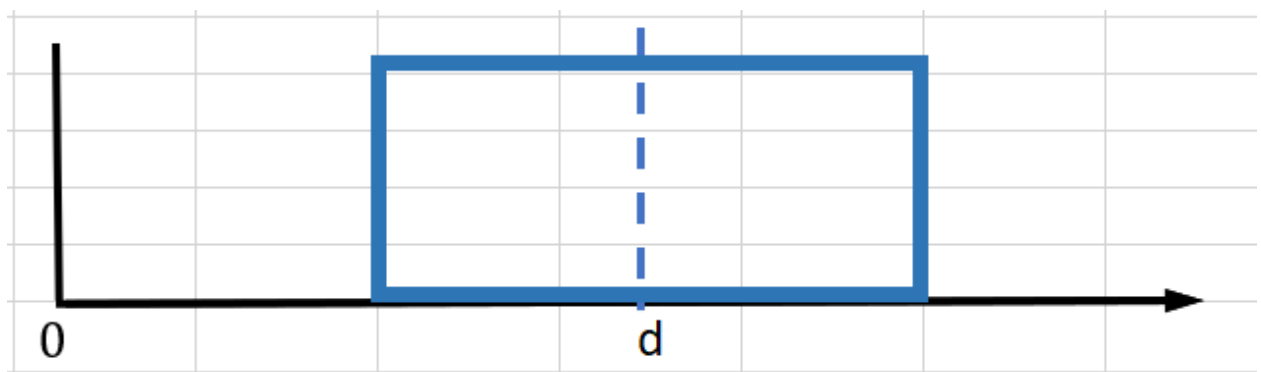


Рисунок 1.1 – Контур розкладу у випадку, коли директивний строк знаходиться далеко від початку координат

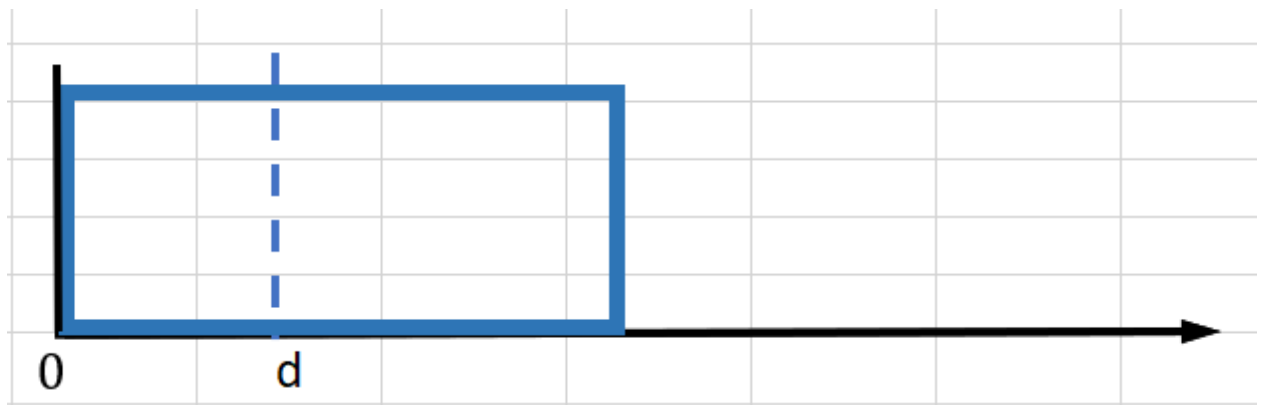


Рисунок 1.2 – Контур розкладу у випадку, коли директивний строк знаходиться близько до початку координат

Для кожного з випадків розроблено свій алгоритм. Їхнє поєднання дає комплексний алгоритм розв'язання задачі з однією машиною, повна схема якого наведена на діаграмі діяльності у графічних матеріалах.

Випадок 1: Директивний строк достатньо великий

У цьому випадку можливо побудувати оптимальний розклад. Ідея полягає в тому, щоб роботи приблизно рівномірно розподілити по обидві сторони від директивного строку, при чому чим довша робота, тим далі вона має знаходитись від директивного строку. Таким чином, тривалість цієї роботи буде врахована в цільову функцію (яку треба мінімізувати) менше разів.

Алгоритм А1

КРОК 1. Впорядкувати роботи за незростанням тривалостей.

КРОК 2. ЦИКЛ по роботах з КРОКОМ 2:

Одну роботу поставити в першу вільну позицію множини W , другу – в останню вільну позицію множини Q .

Приклад 1

Дано:

$$J = \{5, 5, 5, 4, 4, 3, 3, 2, 1\};$$

$$d = 19.$$

На рисунку 1.3 зображено розклад, побудований для даного набору вхідних даних. Значення цільової функції при цьому:

					ДП ІС-6119.00.001 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

$$L = (0 + 1 + 4 + 8 + 13) + (2 + 5 + 9 + 14) = 26 + 30 = 56$$

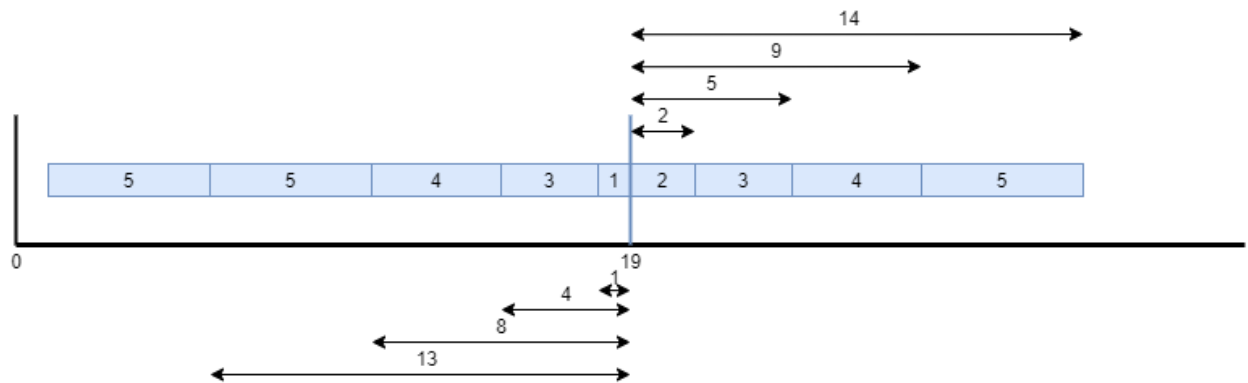


Рисунок 1.3 – Побудований розклад для випадку великого директивного строку

Випадок 2: Директивний строк достатньо малий

У цьому випадку алгоритм А1 не можна застосувати, так як розклад, побудований за ним, починався б раніше, ніж нульова точка, тобто ще до моменту надходження робіт, що неможливо. У такому випадку процес побудови розкладу трохи відрізняється. Ідея алгоритму для цієї ситуації наступна: якщо робота може бути поставлена до директивного строку (часу до нього достатньо), то вона ставиться, якщо ні, то робота ставиться після директивного строку. Знову ж таки, чим робота довша, тим вона далі має знаходитись від директивного строку.

Алгоритм А2

КРОК 1. Впорядкувати роботи за незростанням тривалостей.

КРОК 2. Ініціалізувати $\tau_1 = d, \tau_2 = \sum p - d$.

КРОК 3. ЦИКЛ по роботах:

ЯКЩО $\tau_1 \geq \tau_2$:

Роботу i призначити на першу незайняту позицію множини W і зменшити τ_1 на p_1 .

ІНАКШЕ:

Роботу i призначити на останню незайняту позицію множини Q і зменшити τ_2 на p_1 .

Приклад 2

Дано:

$$J = \{5, 5, 5, 4, 4, 3, 3, 2, 1\};$$

$$d = 11.$$

На рисунку 1.4 зображено розклад, побудований для даної ситуації. Можна побачити, що виконання робіт, на відміну від попереднього випадку, починається одразу після їх надходження, тобто в нульовий момент часу. Окрім того, більшість робіт тепер знаходиться справа від директивного строку. Це відбулося через неможливість виконання даних робіт до директивного строку.

Значення цільової функції при цьому:

$$L = (0 + 1 + 6) + (2 + 5 + 8 + 12 + 16 + 21) = 7 + 64 = 71$$

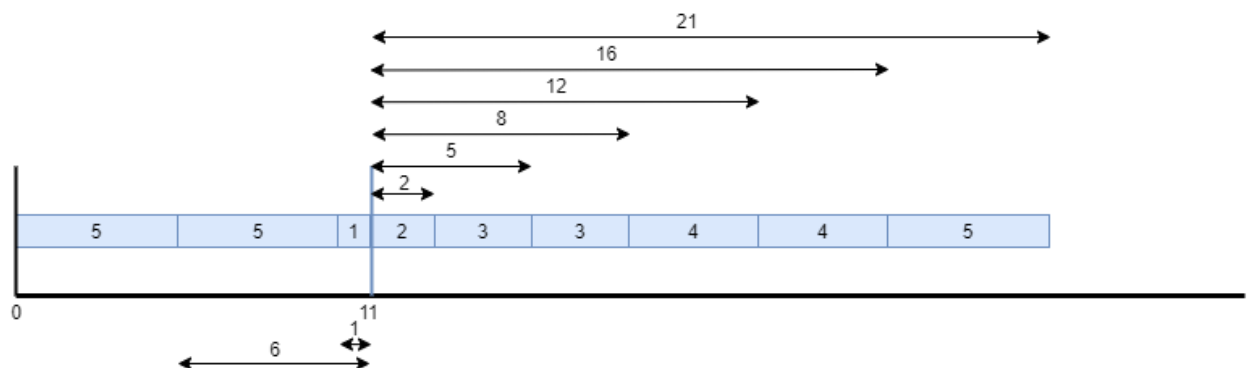


Рисунок 1.4 – Побудований розклад для випадку малого директивного строку

Окрім того, при малому директивному строку може виникнути ще один частковий випадок – випадок «дуже малого» директивного строку. Ця ситуація виникає, коли значення директивного строку менше за тривалість найкоротшої з робіт. Цей випадок, насправді, може бути вирішений алгоритмом А2 при зміщенні побудованого розкладу до початку координат. Проте, для цього випадку вже існує алгоритм – SPT (Shortest Processing Time) [4]. Саме він буде використовуватись для розв'язання задач при названих умовах.

Його ідея полягає в тому, щоб розташовувати роботи в порядку зростання тривалостей, починаючи з початку координат. Таким чином буде мінімізуватись середній час перебування роботи в системі, що, у свою чергу, мінімізуватиме цільову функцію, адже так само, як в алгоритмі А1, найдовші роботи стоятимуть далеко від директивного строку, відповідно, вони будуть входити в цільову функцію меншу кількість разів.

Приклад 3

$$J = \{6, 7, 8, 9\};$$

$$d = 5.$$

На рисунку 1.5 зображено розклад, побудований за алгоритмом SPT. Можна побачити, що роботи стоять в порядку зростання, а сам розклад починається з точки початку координат.

Значення цільової функції при цьому:

$$L = (1 + 8 + 16 + 25) = 50$$

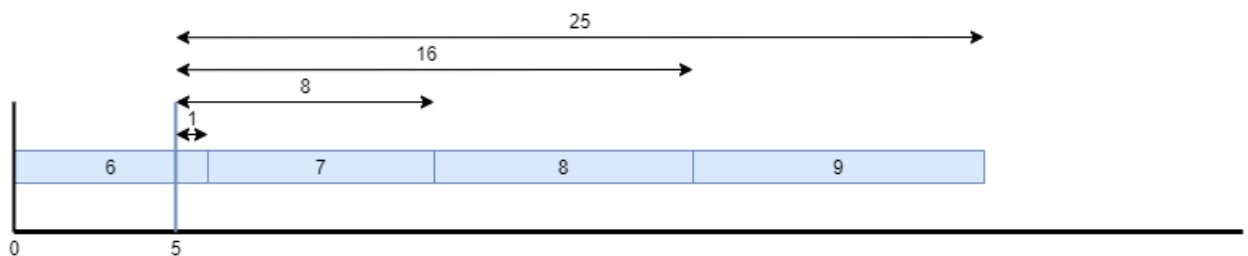


Рисунок 1.5 – Побудований розклад для випадку дуже малого директивного строку

Розроблені алгоритми можна застосувати і для декількох машин. Для цього потрібно знайти спосіб розділення робіт між машинами. Коли до кожної машини буде прив'язана певна множина робіт, дані алгоритми можна буде використати для кожної машини окремо.

Логічним видається розділяти роботи рівномірно між машинами, починаючи з найдовшої роботи і закінчуючи найкоротшою. Проте, у такий спосіб остання машина буде отримувати значно коротші роботи, ніж перша, через що буде виникати дисбаланс.

Тому пропонується наступний «хвильовий» метод: після «видачі» кожній машині по одній роботі, починати наступне коло розділення з останньої машини. Тобто, доходячи до останньої машини, їй буде приписуватись дві роботи, після чого процес буде продовжуватись у зворотному порядку. Таким чином буде відбуватися врівноваження тривалостей робіт, приписаних кожній машині. Детальний приклад розділення робіт між машинами за «хвильовим» алгоритмом наведено у графічних матеріалах.

Приклад 4

$$J = \{3, 4, 5, 5, 6, 8, 9, 11\};$$

$$m = 3.$$

Розподіл наведено в таблиці 1.1. Можна побачити, що відсортовані у порядку спадання роботи розподіляються між машинами наче хвилі. Від цього і назва алгоритму. У результаті вийде розклад, представлений на рисунку 1.6.

Таблиця 1.1 – Приклад розподілу робіт між машинами за «хвильовим» алгоритмом

Тривалість роботи	11	9	8	6	5	5	4	3
Машина	1	2	3	3	2	1	1	2

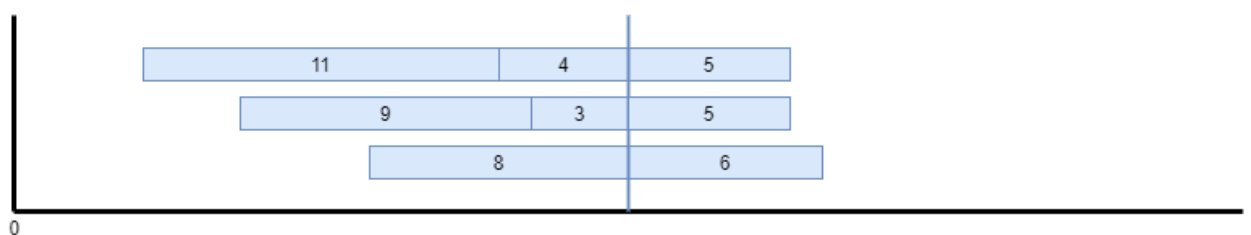


Рисунок 1.6 – Приклад побудованого розкладу після розподілу робіт «хвильовим» алгоритмом

Використовуючи наведений алгоритм, можна розділити роботи між машинами, після чого для кожної окремої машини побудувати розклад за алгоритмами А1, А2.

Приклад 5

$$J = \{3, 4, 5, 5, 6, 8, 9, 11\};$$

$$d = 5;$$

$$m = 3.$$

На рисунку 1.7 зображено побудований розклад для наведених вхідних даних. Це випадок малого директивного строку. Можна побачити, що роботи спочатку було розділено між машинами (таблиця 1.2), а потім визначено їх позиції в розкладі.

Таким чином, за допомогою «хвильового» алгоритму задача з декількома машинами була зведена до розв'язання декількох задач з однією машиною.

Таблиця 1.2 – Приклад розподілу робіт між машинами за «хвильовим» алгоритмом

Тривалість роботи	11	9	8	6	5	5	4	3
Машина	1	2	3	3	2	1	1	2

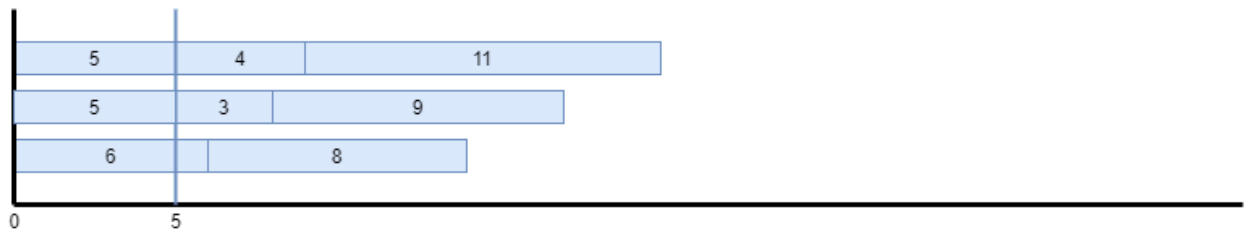


Рисунок 1.7 – Приклад побудованого розкладу після розподілу робіт «хвильовим» алгоритмом для ситуації малого директивного строку

У випадку, якщо машини мають різні продуктивності роботи, до «хвильового» алгоритму додається ще один крок на початок. Перед розподілом робіт потрібно посортувати машини у порядку спадання продуктивностей. Таким чином, найдовші роботи на кожному кроці розподілу будуть передаватися найшвидшій машині, врівноважуючи цим самим загальний контур побудованого розкладу.

Приклад 6

$$J = \{3, 4, 5, 5, 6, 8, 9, 11\};$$

$$m = 3;$$

$$h = \{1.2, 1, 0.8\}.$$

За таких умов, розклад, побудований після розподілу, буде мати трохи інший вигляд (рис. 1.8). Тривалості робіт на кожній з машин будуть відрізнятися від заданих в умові і будуть відповідати значенням, наведеним у таблиці 1.3.

Таблиця 1.3 – Приклад розподілу робіт між машинами з різною продуктивністю

Машина	1	2	3	3	2	1	1	2
Продуктивність	1.2	1	0.8	0.8	1	1.2	1.2	1
Тривалість роботи	11	9	8	6	5	5	4	3
Тривалість на машині	9.167	9	10	7.5	5	4.167	3.33	3

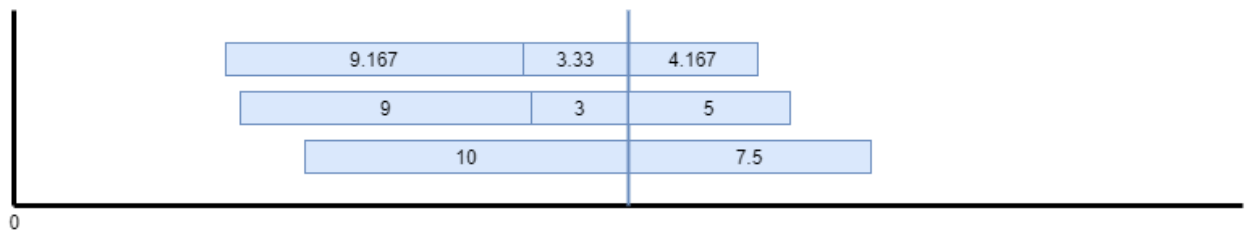


Рисунок 1.8 – Приклад побудованого розкладу після розподілу робіт «хвильовим» алгоритмом для машин з різною продуктивністю

Можна побачити, що впорядкування робіт в розкладі не змінилося, проте, сам вигляд розкладу став компактнішим. І хоча розклад, побудований у даному випадку, може і не буде оптимальним, при збільшенні кількості робіт і унеможливленні повного перебору даний алгоритм буде давати хороші результати за прийнятний час.

Висновок до розділу

У даному розділі було наведено індивідуальну частину математичного забезпечення дипломного проекту. Було розглянуто задачі з одним директивним строком – один із класів задач теорії розкладів, який

розглядається в роботі. Було наведено постановку задачі, часткові випадки, що виникають при її розв'язанні, алгоритми для побудови розкладів, наведено приклади їх роботи. Окрім цього, було описано «хвильовий» алгоритм – спосіб розподілу робіт між машинами, який використовується для розбиття задач з декількома машинами на підзадачі з одною машиною, які розв'язуються за допомогою описаних в роботі алгоритмів.

					ДП ІС-6119.00.001 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Архітектура програмного забезпечення

2.1.1 Специфікація функцій

У даній індивідуальній частині були розроблені деякі компоненти графічного інтерфейсу, деякий функціонал класу Schedule, а також підсистеми візуалізації результатів та генерації звітності, що представлені за допомогою класу Gantt та деяких методів класу Research. У таблицях 2.1-2.5 наведено опис функцій, розроблених для вищевказаного функціоналу.

Таблиця 2.1 – Опис методів класу ResearchOneDueDate

Метод	Опис методу	Параметри	Опис параметрів
__init__	Конструктор класу, створює вікно	parent, controller	Батьківський віджет, контролер
research	Викликає об'єкт класу Research для проведення дослідження алгоритму	-	-

Таблиця 2.2 – Опис деяких методів класу Schedule

Метод	Опис методу	Параметри	Опис параметрів
spt	Будує розклад для випадку, коли директивний строк менший за найкоротшу роботу	jobs, machine, start	Список тривалостей робіт, продуктивність машини, для якої будується розклад, час можливого початку виконання робіт

Продовження таблиці 2.2

Метод	Опис методу	Параметри	Опис параметрів
tight_due_date	Будує розклад для випадку, коли директивний строк достатньо близько до початку координат	jobs, machine, d, start	Список тривалостей робіт, продуктивність машини, для якої будується розклад, значення директивного строку час можливого початку виконання робіт
loose_due_date	Будує розклад для випадку, коли директивний строк достатньо великий	jobs, d, machine	Список тривалостей робіт, значення директивного строку, продуктивність машини, для якої будується розклад
divide	Розділяє роботи між машинами	num_d, jobs, num_machines	Кількість директивних строків, список тривалостей робіт, кількість машин
one_due_date	Визначає потрібний алгоритм для побудови розкладу у задачі з одним директивним строком	jobs, d, num_machines, machines	Список тривалостей робіт, значення директивного строку, кількість машин, список продуктивностей машин

Продовження таблиці 2.2

Метод	Опис методу	Параметри	Опис параметрів
decide	Визначає тип задачі відповідно до кількості директивних строків	num_d, jobs, d, num_machines, machines	Кількість директивних строків, список тривалостей робіт, список значень директивних строків, кількість машин, список продуктивностей машин

Таблиця 2.3 – Опис деяких методів класу Research

Метод	Опис методу	Параметри	Опис параметрів
compare	Будує графік дослідження алгоритмів	z_arr, perfect_arr	Значення цільових функцій побудованих розкладів, значення цільових функцій для ідеального (неможливого) випадку

Таблиця 2.4 – Опис методів класу Test

Метод	Опис методу	Параметри	Опис параметрів
__init__	Конструктор класу, ініціалізує змінні	parent, controller	Батьківський віджет, контролер
get_filename	Надає можливість обрати файл з вхідними даними	-	-
call_graph	Викликає об'єкт класу Gantt для побудови візуалізації розкладу	-	-

Таблиця 2.5 – Опис методів класу Gantt

Метод	Опис методу	Параметри	Опис параметрів
__init__	Конструктор класу, ініціалізує змінні	path	Шлях до файлу з вхідними даними
get_schedule	Викликає об'єкт класу Schedule для побудови розкладу	-	-
draw_graph	Будує графік з візуалізацією розкладу	-	-

2.2 Опис звітів

Як було вже сказано у загальній частині дипломного проекту, розроблена програма може працювати у двох різних режимах. При роботі у режимі тестування алгоритму результатом виконання є екранна форма з побудованим розкладом (рис. 2.1), а при дослідженні алгоритмів – екранна форма з результатом порівняння значень цільової функції побудованих розкладів для згенерованих вхідних даних та для їхніх нижніх границь (рис.2.2).

У даній індивідуальній частині йдеться про дослідження задач з одним директивним строком. Результат виконання програми у режимі дослідження алгоритмів розв'язання задач з двома директивними строками є схожим до наведеного у даній частині. Приклад звіту цього процесу представлено у індивідуальній частині № 2.

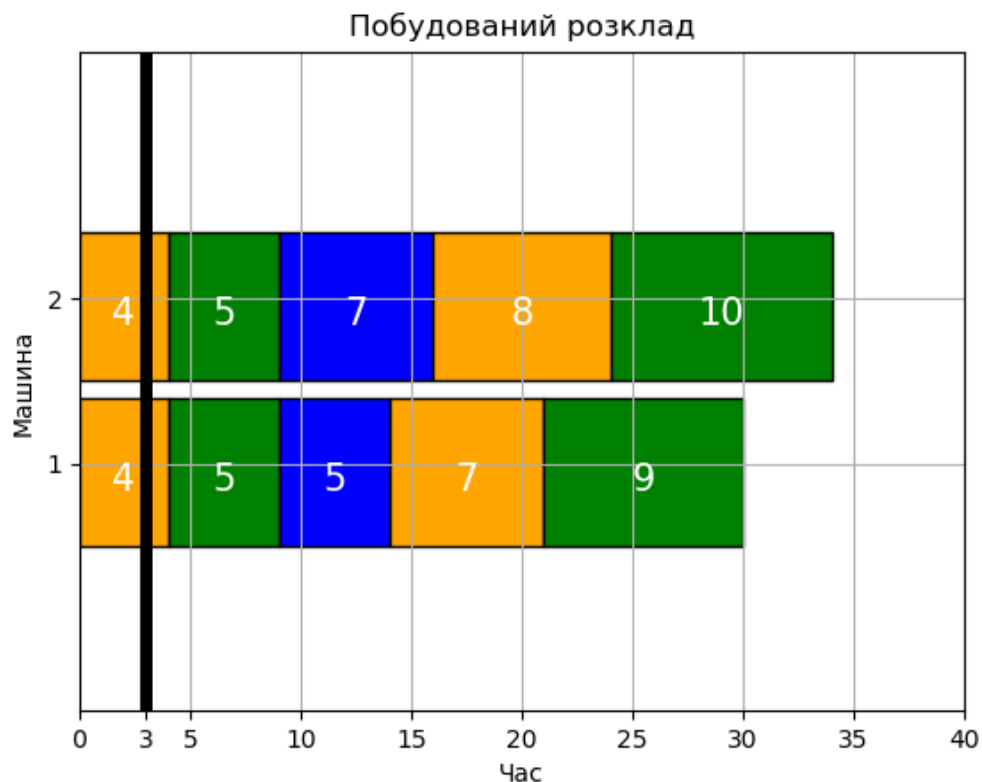


Рисунок 2.1 – Приклад екранної форми для візуалізації побудованого розкладу

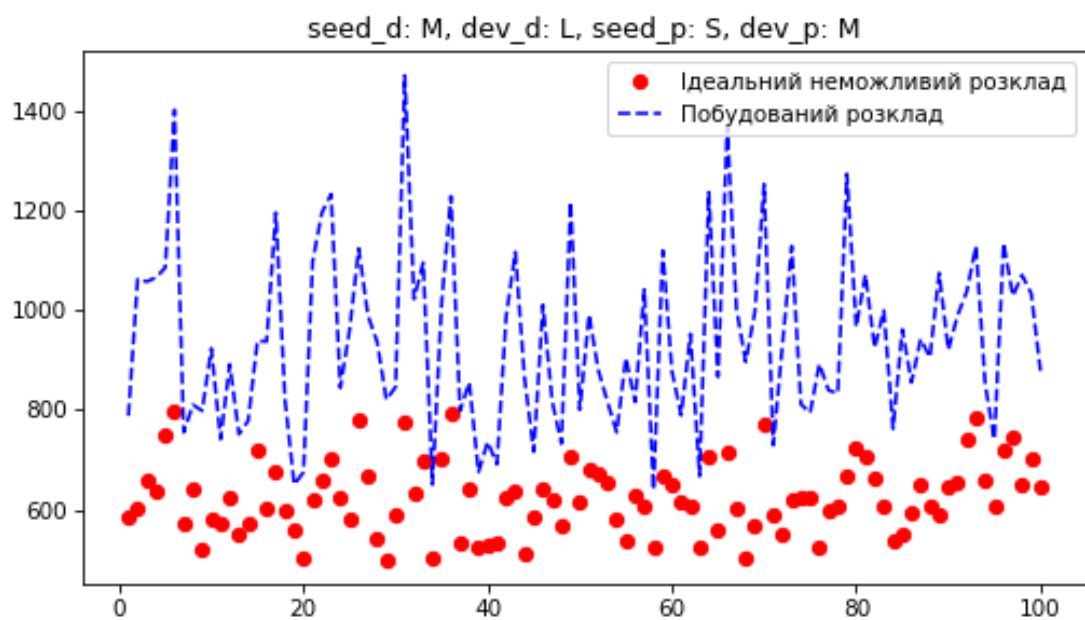


Рисунок 2.2 – Приклад екранної форми з результатами дослідження алгоритмів

Висновок до розділу

У даному розділі було описано деякий функціонал програмного продукту, а саме – наведено детальний опис методів класів, що відповідають за частину функціоналу, описаного в розділі математичного забезпечення. Було наведено приклади звітностей, згенерованих програмою, які відображають результати роботи продукту в обох режимах.

					ДП ІС-6119.00.001 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

3 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

3.1 Керівництво користувача

Головне вікно програмного продукту (рис. 3.1), що наведене у загальній частині дипломної роботи, є першим, що бачить користувач, починаючи роботу з системою. На даному вікні можна обрати одну з опцій в залежності від того, що саме хоче зробити користувач.

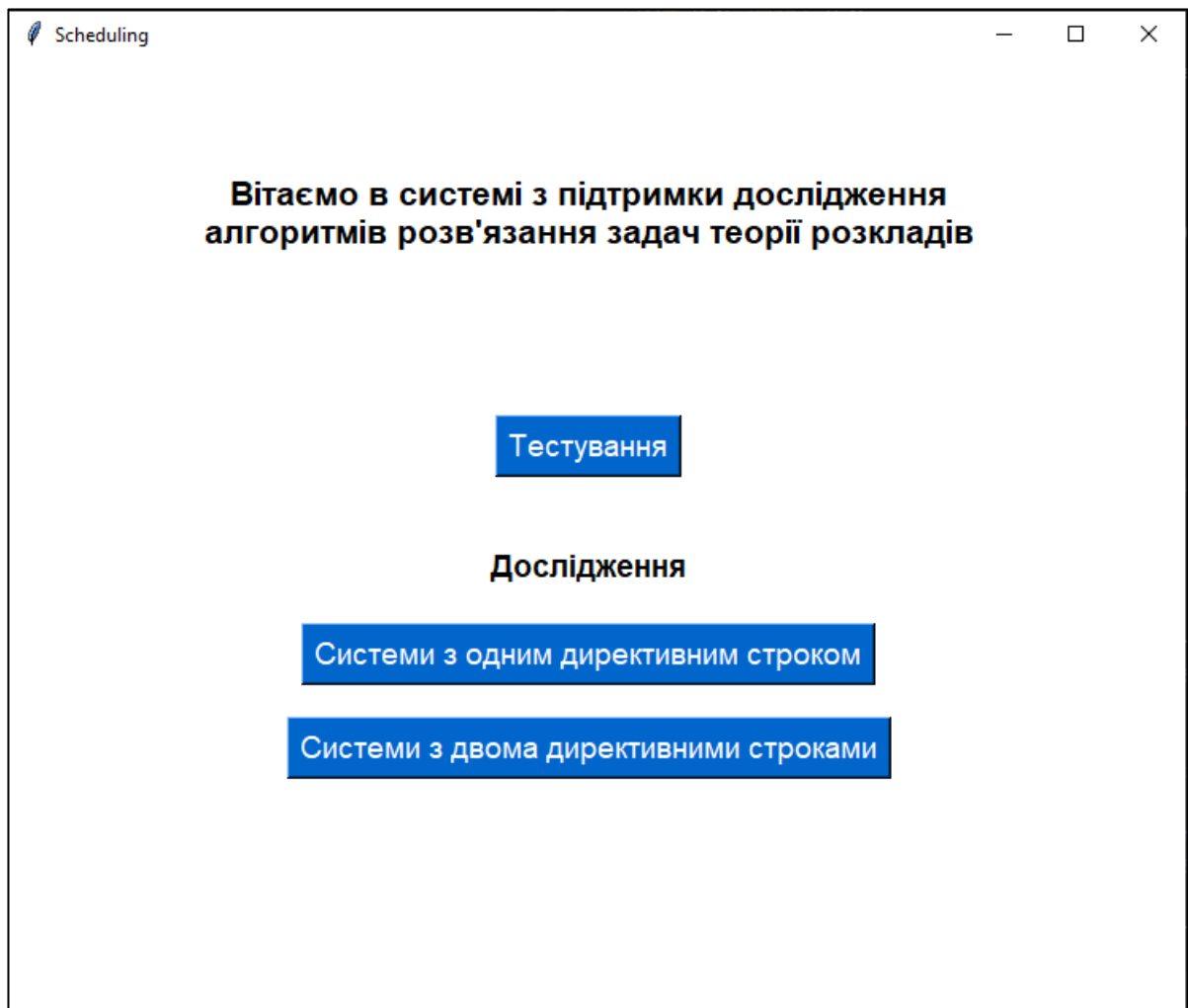


Рисунок 3.1 – Головне вікно програми

У разі, якщо він бажає дослідити роботу певних алгоритмів для своїх вхідних даних, потрібно натиснути кнопку «Тестування», після чого відкриється наступне вікно, на якому можна обрати файл з вхідними даними (рис. 3.2).

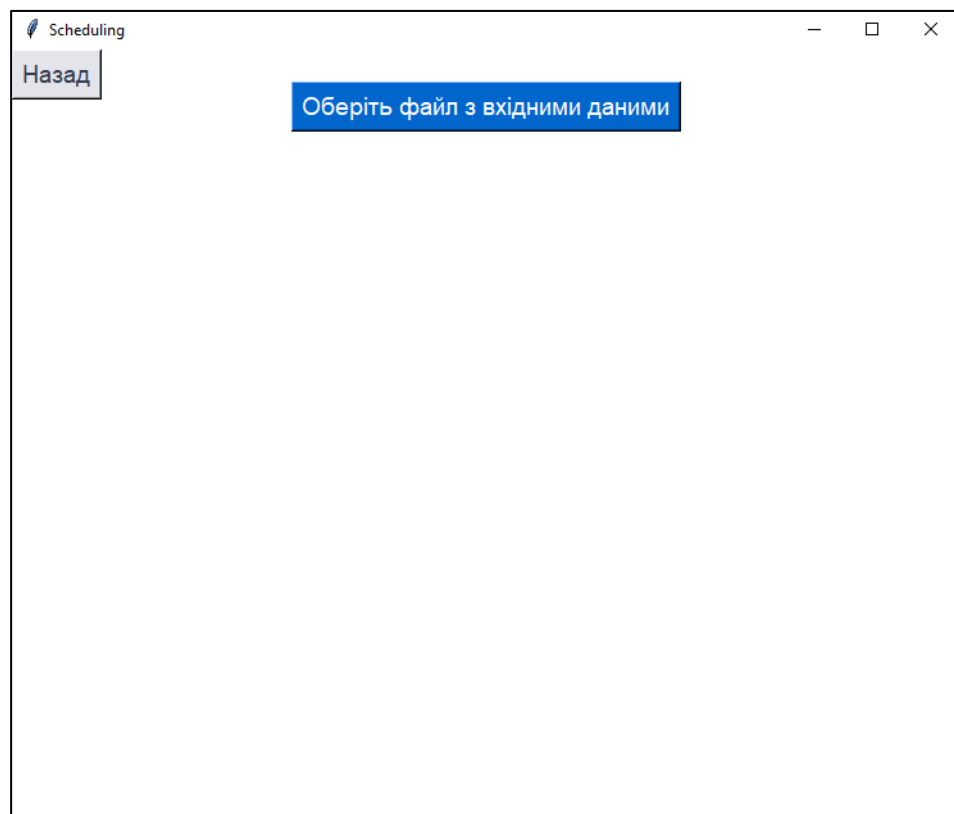


Рисунок 3.2 – Екранна форма для вибору файлу з вхідними даними

На даній екранній формі потрібно натиснути кнопку «Оберіть файл з вхідними даними», після чого з'явиться стандартний діалог, який дозволяє оглянути наявні файли у файловій системі локального комп'ютера та обрати потрібний (рис. 3.3).

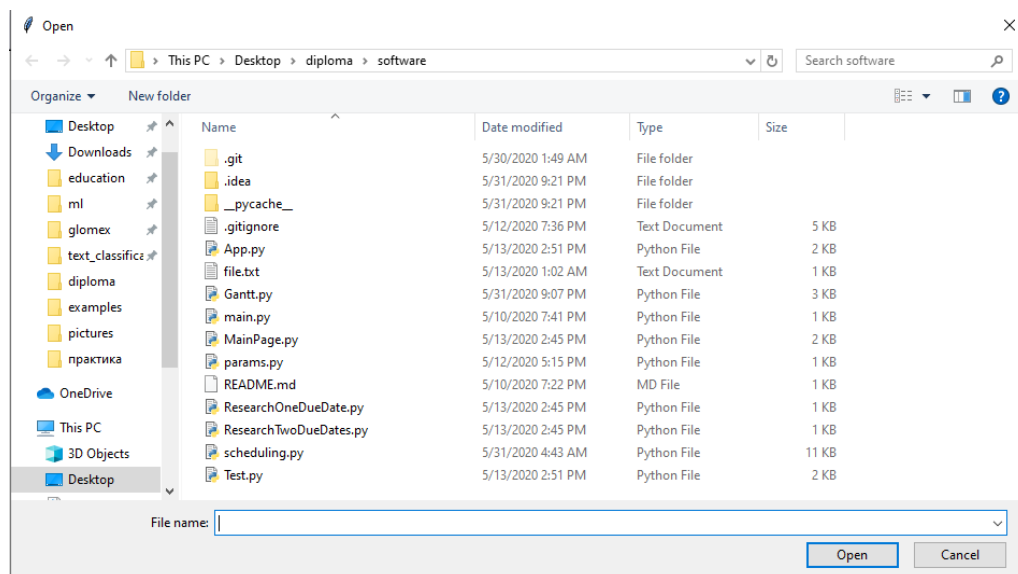


Рисунок 3.3 – Стандартне діалогове вікно Windows для вибору файлу з файлової системи

Обравши необхідний файл, шлях до нього з'явиться на екранній формі (рис. 3.4).

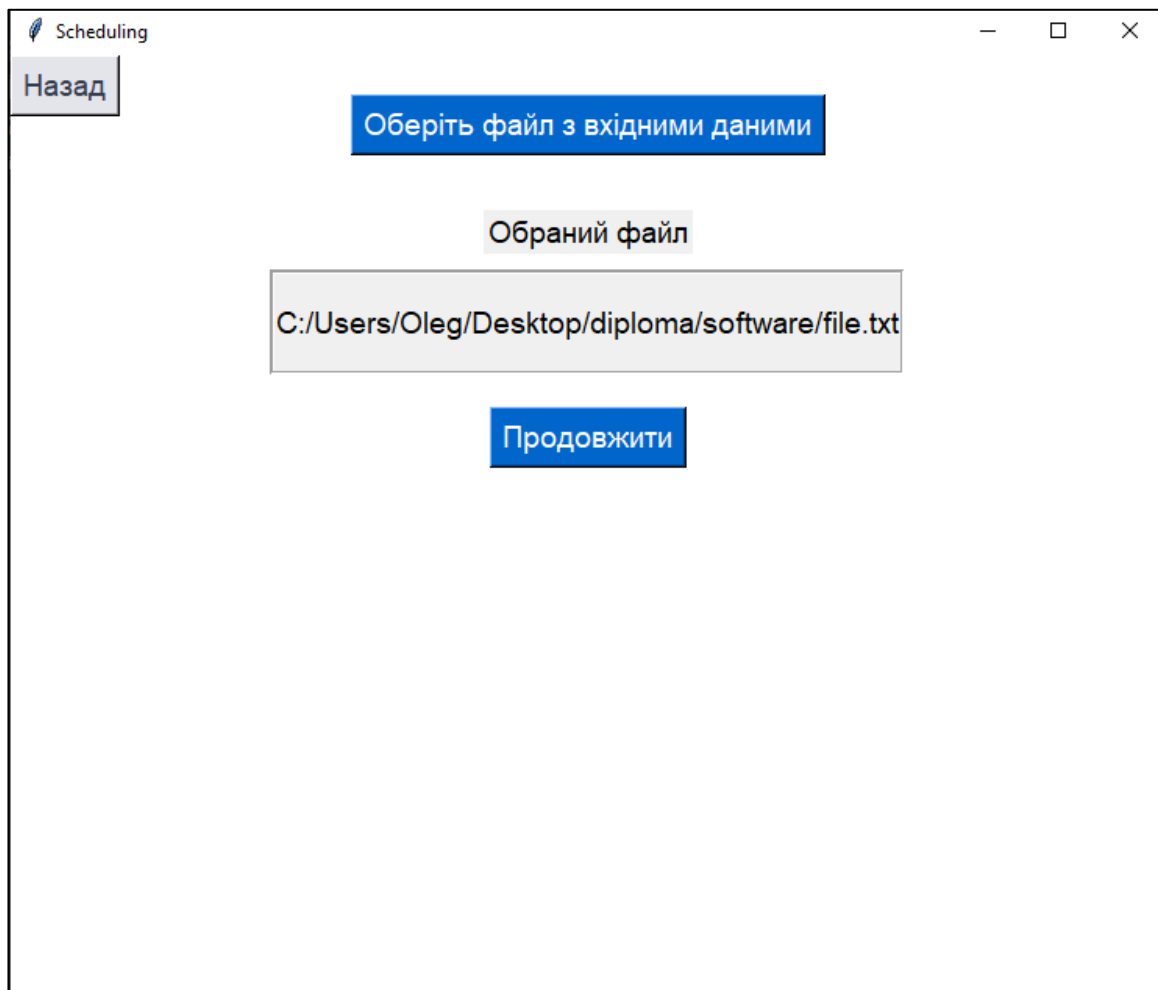


Рисунок 3.4 – Екранна форма зі шляхом до обраного файлу

Користувач може перевірити правильність обраного шляху, після чого натискає «Продовжити». Нарешті, з'явиться екранна форма з побудованим розкладом (рис. 3.5). Окрім того, у ту ж директорії, з якої було взято файл з вхідними даними, записується новий файл з назвою result.txt, в якому можна знайти результат побудови розкладу (рис. 3.6). Формат даного файлу описано в розділі інформаційного забезпечення загальної частини дипломного проекту.

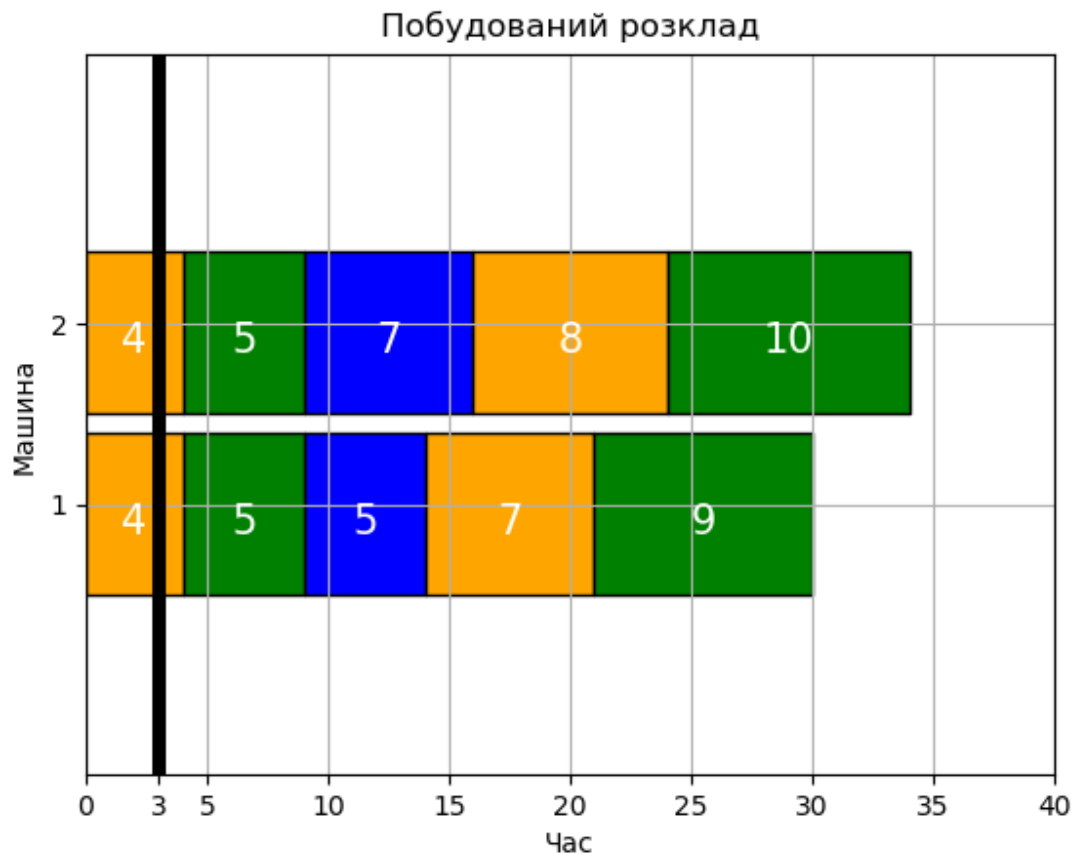


Рисунок 3.5 – Екранна форма з візуалізацією побудованого розкладу

1	20	30
2	1	
3	6,10	16,4 20,5
4	25,5	
5	1	
6	8,9	17,3 20,6
7	26,4	30,1
8	1	
9	12,8	20,7
10	27,3	30,2

Рисунок 3.6 – Приклад файлу з результатами побудованого розкладу

У випадку, коли користувач бажає дослідити роботу алгоритмів розв'язання задач з одним директивним строком, він на головному екрані обирає опцію «Системи з одним директивним строком». Відкриється вікно для введення параметрів генератора вхідних даних (рис. 3.7).

Scheduling

Назад

Параметри генератора

Кількість робіт S

Розмір директивного строку S

Розміри робіт S

Кількість машин S

Продуктивності машин S

Продовжити

Повне дослідження

Рисунок 3.7 – Екранна форма для введення параметрів генератора

Введення параметрів є необов'язковою дією і призначене для випадків, коли користувач хоче дослідити лише деякі випадки задачі. Якщо дослідник хоче провести повне дослідження, потрібно обрати опцію «Повне дослідження». Наступною екранною формою буде результат дослідження – графік з порівняннями цільових функцій (детальніше про це в індивідуальній частині № 2). Приклад такого графіку наведено на рисунку 3.8.

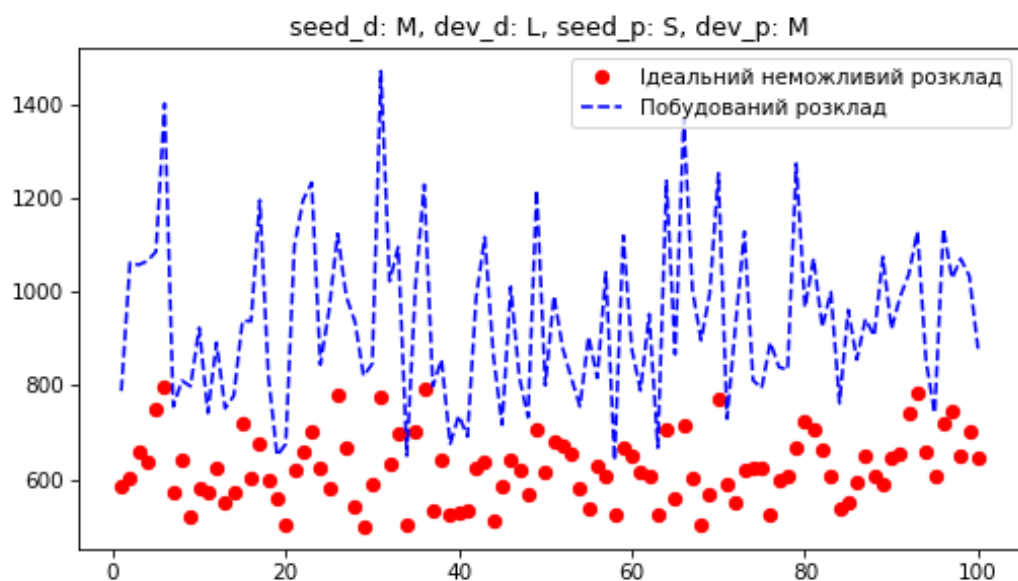


Рисунок 3.8 – Екранна форма з результатами дослідження

Змн.	Арк.	№ докум.	Підпис	Дата

3.2 Випробування програмного продукту

3.2.1 Мета випробувань

Метою випробувань є перевірка коректності розроблених алгоритмів, а також результатів дослідження алгоритмів.

3.2.2 Загальні положення

Для перевірки коректності роботи алгоритмів використовується стандартний «ручний» метод: для заданих вхідних даних будується розклад «вручну», після чого він звіряється з розкладом, побудованим програмним продуктом.

Результати дослідження алгоритмів дають можливість лише сказати, наскільки розроблені алгоритми є ефективними для розв'язання задач теорії розкладів.

3.2.3 Результати випробувань

Рисунки 3.9-3.13 демонструють результати роботи програмного продукту для вхідних даних, використаних для прикладів у розділі математичного забезпечення. Порівнюючи розклади, побудовані вручну в математичному розділі та програмно в даному розділі, можна побачити, що вони відповідають одне одному, що свідчить про коректність роботи алгоритмів.

Приклади результатів дослідження алгоритмів розв'язання задач з одним директивним строком при різних значеннях параметрів генератора наведено в графічних матеріалах.

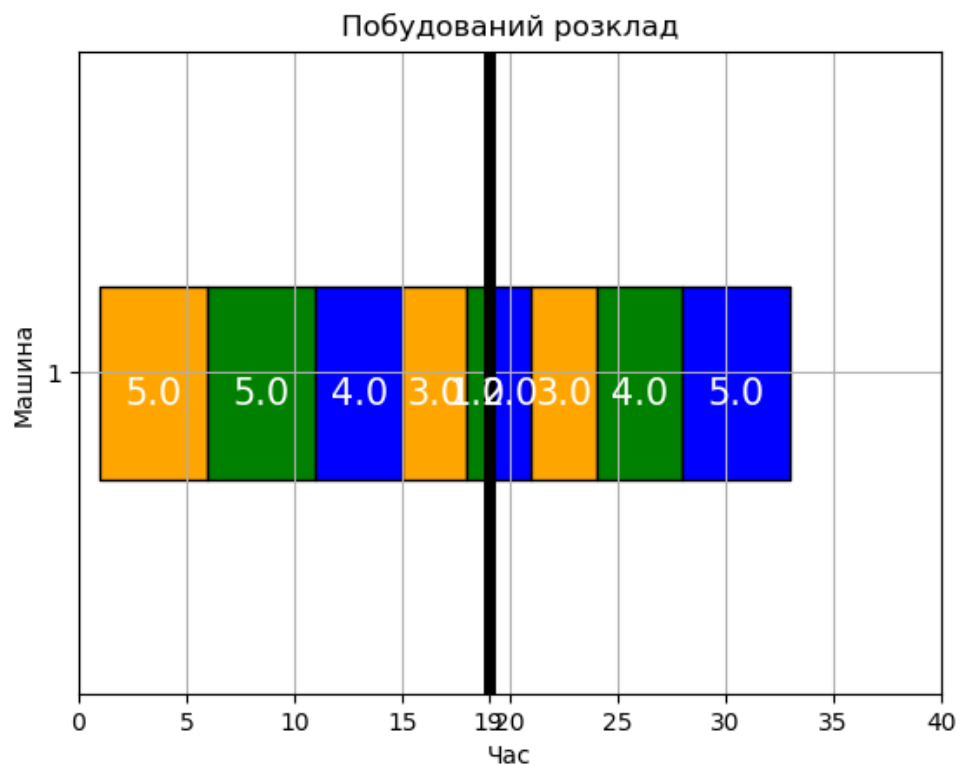


Рисунок 3.9 – Побудований розклад для вхідних даних прикладу 1

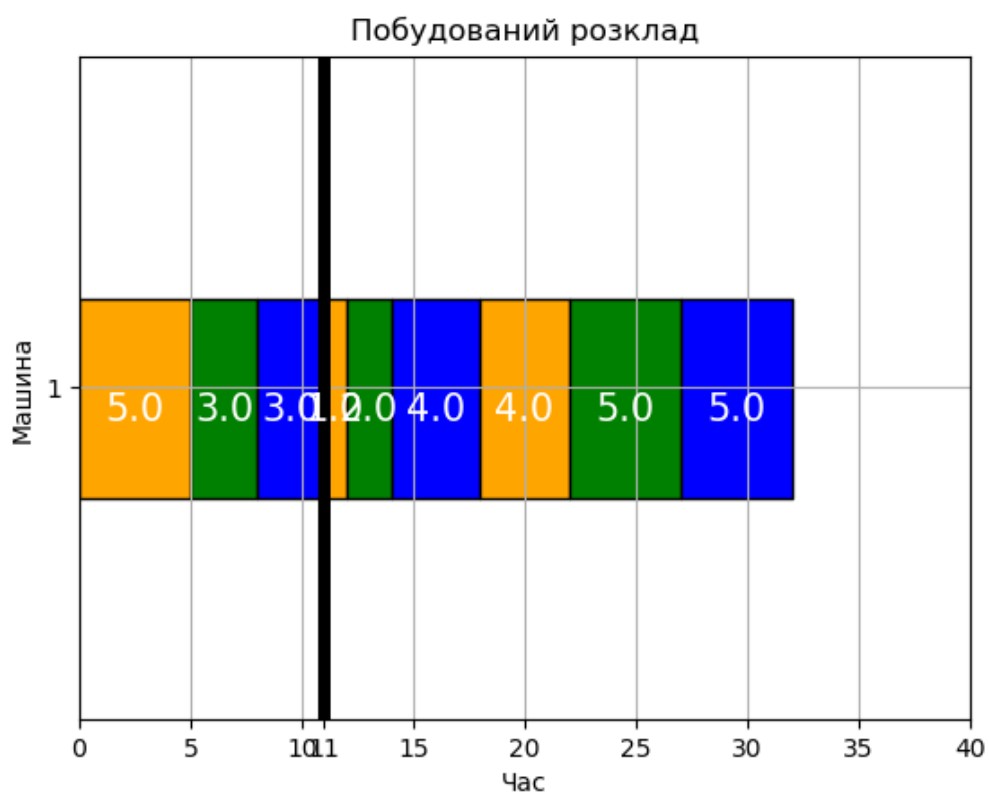


Рисунок 3.10 – Побудований розклад для вхідних даних прикладу 2

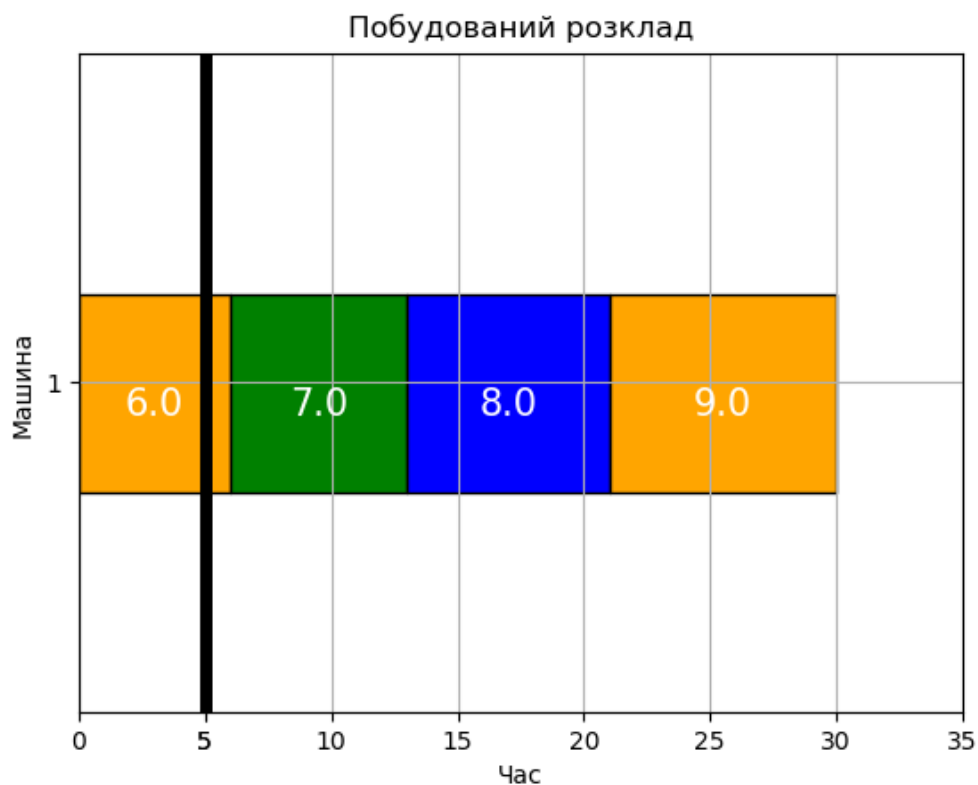


Рисунок 3.11 – Побудований розклад для вхідних даних прикладу 3

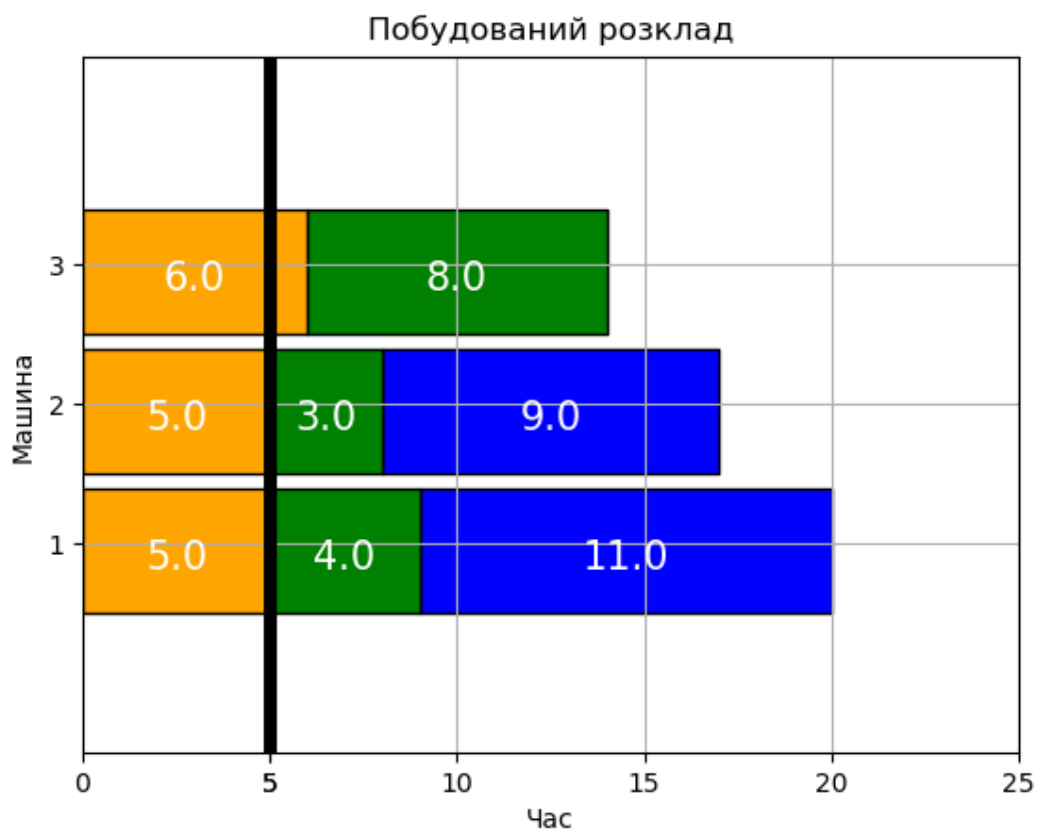


Рисунок 3.12 – Побудований розклад для вхідних даних прикладу 5

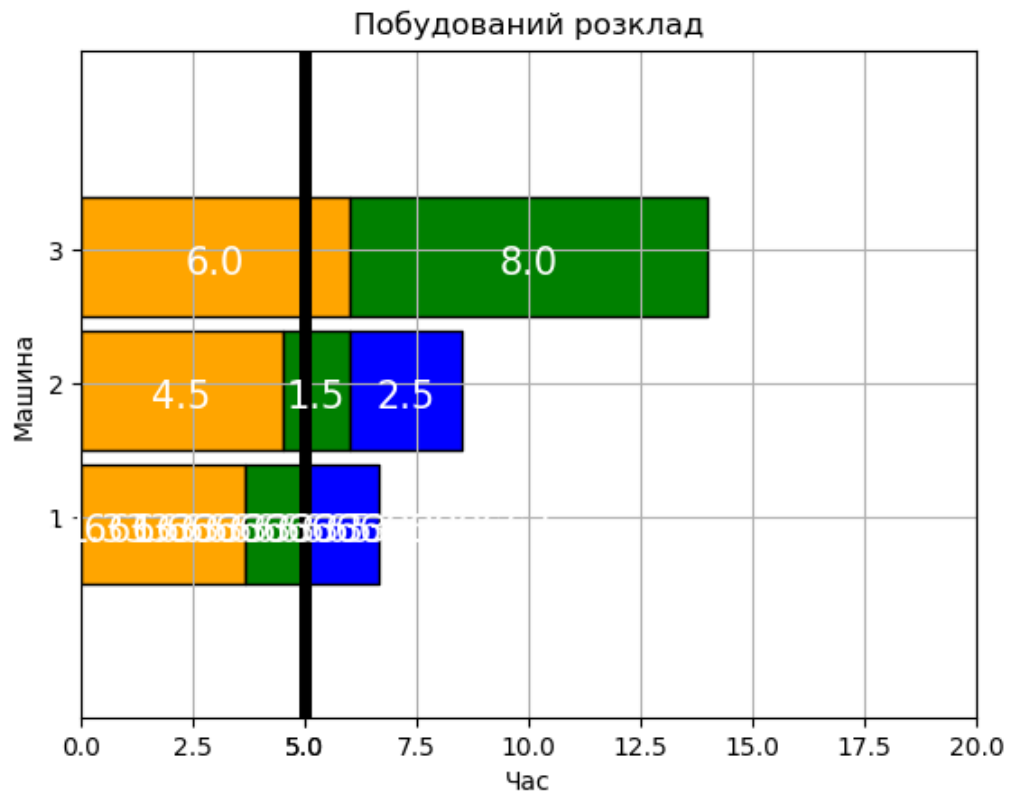


Рисунок 3.13 – Побудований розклад для вхідних даних прикладу 6

Висновок до розділу

У даному розділі було представлено керівництво користувача для випадків, коли дослідник бажає протестувати алгоритм або дослідити клас задач з одним директивним строком. Було продемонстровано покрокові інструкції з екранними формами, які демонструють процес роботи з програмою.

Окрім того, наведено детальне випробування продукту, перевірено правильність побудови розкладів та наведено приклади результатів дослідження.

ЗАГАЛЬНІ ВИСНОВКИ

Дана індивідуальна частина присвячена розробці алгоритмів розв'язання задач теорії розкладів з одним директивним строком, їх розширенню для умов із декількома машинами, а також програмній реалізації даних алгоритмів та підсистем візуалізації та Було описано та продемонстровано на прикладах роботу реалізованих алгоритмів, які склали основу для їх подальшого використання при роботі із задачами з двома директивними строками (які описуються у індивідуальній частині № 2).

У розділі програмного та технічного забезпечення наведено опис методів, реалізованих відповідно до математичної частини програмного продукту.

Розділ технічного забезпечення надає посібник користувача для випадків тестування алгоритмів та дослідження задач з одним директивним строком, а також наводить приклади випробування продукту.

ПЕРЕЛІК ПОСИЛАНЬ

1. Peter Brucker. Scheduling Algorithms / Peter Brucker. – New York : Springer, 2007. – 378 p.
2. J.A. Hoogeveen, S.L. van de Velde. Scheduling around a small common due date / J.A. Hoogeveen, S.L. van de Velde // European Journal of Operational Research. – 1991. – № 55. – pp. 237-242.
3. Michael L. Pinedo. Scheduling: Theory, Algorithms, and Systems / Michael L. Pinedo. – New York : Springer, 2008. – 662 p.
4. (2000) SHORTEST PROCESSING TIME RULE. In: Swamidass P.M. (eds) Encyclopedia of Production and Manufacturing Management. Springer, Boston, MA.

ДОДАТОК А

Тексти програмного коду

*Система з підтримки процесу дослідження алгоритмів розв'язання
деякого класу задач теорії розкладів. Задачі з одним директивним
строком. Підсистеми візуалізації та аналізу результатів*

(Найменування програми (документа))

DVD-R

(Вид носія даних)

40 арк. 465 КБ

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2020 року

					ДП ІС-6119.00.001 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

```

from params import *
import Gantt

class ResearchOneDueDate(tkinter.Frame):
    def __init__(self, parent, controller):
        tkinter.Frame.__init__(self, parent)
        self.controller = controller

        arr = ['S', 'M', 'L']

        self.button = tkinter.Button(self, text='Назад', font=font,
bg=light_grey_color, fg=dark_grey_color,
                                command=lambda: self.reset_frame()).grid(row=0,
columnspan=2, sticky='W')

        tkinter.Label(self, text='Параметри генератора',
font=font_header).grid(row=1, columnspan=2, padx=275, pady=50)

        tkinter.Label(self, text='Кількість робіт', font=font).grid(row=2, column=0,
padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=2, column=1, padx=10, pady=10, sticky='w')

        tkinter.Label(self, text='Розмір директивного строку', font=font).grid(row=3,
column=0, padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=3, column=1, padx=10, pady=10, sticky='w')

        tkinter.Label(self, text='Розміри робіт', font=font).grid(row=4, column=0,
padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=4, column=1, padx=10, pady=10, sticky='w')

        tkinter.Label(self, text='Кількість машин', font=font).grid(row=5, column=0,
padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=5, column=1, padx=10, pady=10, sticky='w')

        tkinter.Label(self, text='Продуктивності машин', font=font).grid(row=6,
column=0, padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=6, column=1, padx=10, pady=10, sticky='w')

        tkinter.Button(self, text='Продовжити', font=font_bold,

```

					ДП ІС-6119.00.001 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        command=lambda: self.kek().grid(row=7, column=0, pady=50)
tkinter.Button(self, text='Повне дослідження', font=font_bold,
               command=lambda: self.kek().grid(row=7, column=1, pady=50,
sticky='w')

def reset_frame(self):
    from MainPage import MainPage
    self.controller.frames[MainPage].tkraise()

def kek(self):
    print('kek')

from params import *
from tkinter.filedialog import askopenfilename
from Gantt import Gantt

class Test(tkinter.Frame):
    def __init__(self, parent, controller):
        tkinter.Frame.__init__(self, parent)
        self.controller = controller

        self.button = tkinter.Button(self, text='Назад', font=font,
bg=light_grey_color, fg=dark_grey_color,
                                command=lambda: self.reset_frame()).place(x=0,
y=0)

        tkinter.Button(self, text='Оберіть файл з вхідними даними', font=font,
bg=blue_color, fg='white',
                                command=lambda: self.get_filename()).pack(pady=25)

        self.literal = tkinter.Label(self, text='Обраний файл', font=font)

        self.text = tkinter.StringVar()
        self.text.set('')
        self.label = tkinter.Label(self, textvariable=self.text, font=font,
borderwidth=3, relief="groove", pady=20)

        self.button = tkinter.Button(self, text='Продовжити', font=font,
bg=blue_color, fg='white',
                                command=lambda: self.call_graph())

    def get_filename(self):
        filename = askopenfilename()

        self.literal.pack(pady=10)
        self.label.pack()
        self.text.set(filename)
        self.button.pack(pady=20)

    def reset_frame(self):
        from MainPage import MainPage
        self.controller.frames[MainPage].tkraise()

    def call_graph(self):

```

```

gantt = Gantt(self.text.get())
gantt.draw_graph()

import matplotlib.pyplot as plt
from Schedule import Schedule

class Gantt:
    def __init__(self, path):
        self.filename = path

    with open(self.filename, 'r') as f:
        self.num_d = int(f.readline())
        self.num_jobs = []
        self.jobs = []
        self.d = []

        for _ in range(self.num_d):
            self.num_jobs.append(int(f.readline()))
            self.jobs.append([int(job) for job in f.readline().split()])
            self.d.append(int(f.readline()))
        self.num_machines = int(f.readline())
        self.machines = [int(machine) for machine in f.readline().split()]

    def get_schedule(self):
        return Schedule().build_schedule(self.num_d, self.d, self.jobs,
self.num_machines, self.machines)

    def draw_graph(self):
        schedule = self.get_schedule()

        fig, gnt = plt.subplots()
        gnt.set_title('Побудований розклад')

        gnt.set_ylim(0, (self.num_machines + 1) * 10 + 10)
        gnt.set_xlim(0, max([job[0] + job[1] for machine in schedule for due_date in
machine for job in due_date]) + 5)

        gnt.set_xlabel('Час')
        gnt.set_ylabel('Машина')

        gnt.grid(True)

        yticks = [(i + 1) * 10 + 5 for i in range(self.num_machines)]
        yticklabels = [str(i + 1) for i in range(self.num_machines)]
        gnt.set_yticks(yticks)
        gnt.set_yticklabels(yticklabels)

        # setting the schedules
        for i in range(self.num_machines):
            for j in range(2):
                gnt.broken_barh(schedule[i][j], ((i + 1) * 10, 9),
                                facecolors=('orange', 'green', 'blue') *
len(self.num_jobs), # mark each job with diff color
                                edgecolors='black') # color of borders
            for due_date in schedule[i]:

```

```

        for job in due_date:
            gnt.text(x=job[0] + job[1] / 2, # x = gorizontal position of the
Label (center of the bar)
                    y=(i + 1) * 10 + 4, # y = vertical position of the
Label
                    s=job[1], # text (Length of the job)
                    ha='center', va='center', color='white', size=15)

# setting the due dates
for i in range(self.num_d):
    gnt.axvline(x=self.d[i], c='black', linewidth=5)
    # gnt.text(x=self.d[i] + 0.5, y=1, s=f'd = {self.d[i]}', color='black',
size=10) # possible way to

# set xtick for due date

gnt.set_xticks(list(gnt.get_xticks()) + self.d) # set xtick for due date

fig.show()

```

```

from copy import deepcopy

```

```

class Schedule:
    def __init__(self):
        pass

    # start - zero point (moment of time which indicates the earliest possible
beginning of schedule)
    def spt(self, jobs, machine=1, start=0):
        jobs = [job/machine for job in sorted(jobs)]

        schedule = []

        for job in jobs:
            schedule.append([start, job])
            start += job

        return schedule

    def tight_due_date(self, jobs, d, machine=1, start=0):
        jobs = [job/machine for job in sorted(jobs, reverse=True)]

        schedule_before_due_date = []
        schedule_after_due_date = []

        tay1 = d
        tay2 = sum(jobs) - d

        for job in jobs:
            if tay1 >= tay2:
                schedule_before_due_date.append([start, job])
                start += job
                tay1 -= job
            else:
                schedule_after_due_date.insert(0, [0, job]) # setting temporary time

```

of start to 0

tay2 -= job

setting times when the job is starting

for job **in** schedule_after_due_date:

job[0] = start

start += job[1]

return schedule_before_due_date + schedule_after_due_date

def loose_due_date(**self**, jobs, d, machine=1):

jobs = [job/machine **for** job **in** sorted(jobs, **reverse**=**True**)]

schedule_before_due_date = [[0, job] **for** job **in** jobs[::2]]

schedule_after_due_date = [[0, job] **for** job **in** jobs[1::2]][::-1] *# [::-1] is used to reverse the list*

duration = **sum**([job[1] **for** job **in** schedule_before_due_date]) *# sum of all jobs that go before due date*

for job **in** schedule_before_due_date:

job[0] = d - duration

duration -= job[1]

at this point duration must be equal to 0

duration = d

for job **in** schedule_after_due_date:

job[0] = duration

duration += job[1]

return schedule_before_due_date + schedule_after_due_date

Власник документу:
Попенко Володимир Дмитрович

ID перевірки:
1003783675

Дата перевірки:
04.06.2020 18:31:29 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
04.06.2020 20:52:31 EEST

ID користувача:
77149

Назва документу: Ivanitsky_bachelor_complex

ID файлу: 1003798156 Кількість сторінок: 32 Кількість слів: 5186 Кількість символів: 36241 Розмір файлу: 323.91 KB

13.6% Схожість

Найбільша схожість: 5.23% з джерело бібліотеки. ID файлу: 1003798153

7.44% Схожість з Інтернет джерелами

100

Page 34

12.5% Текстові збіги по Бібліотеці акаунту

186

Page 34

0.15% Цитат

Цитати

1

Page 35

Вилучення переліку посилань вимкнено

0% Вилучень

Вилучений текст відсутній

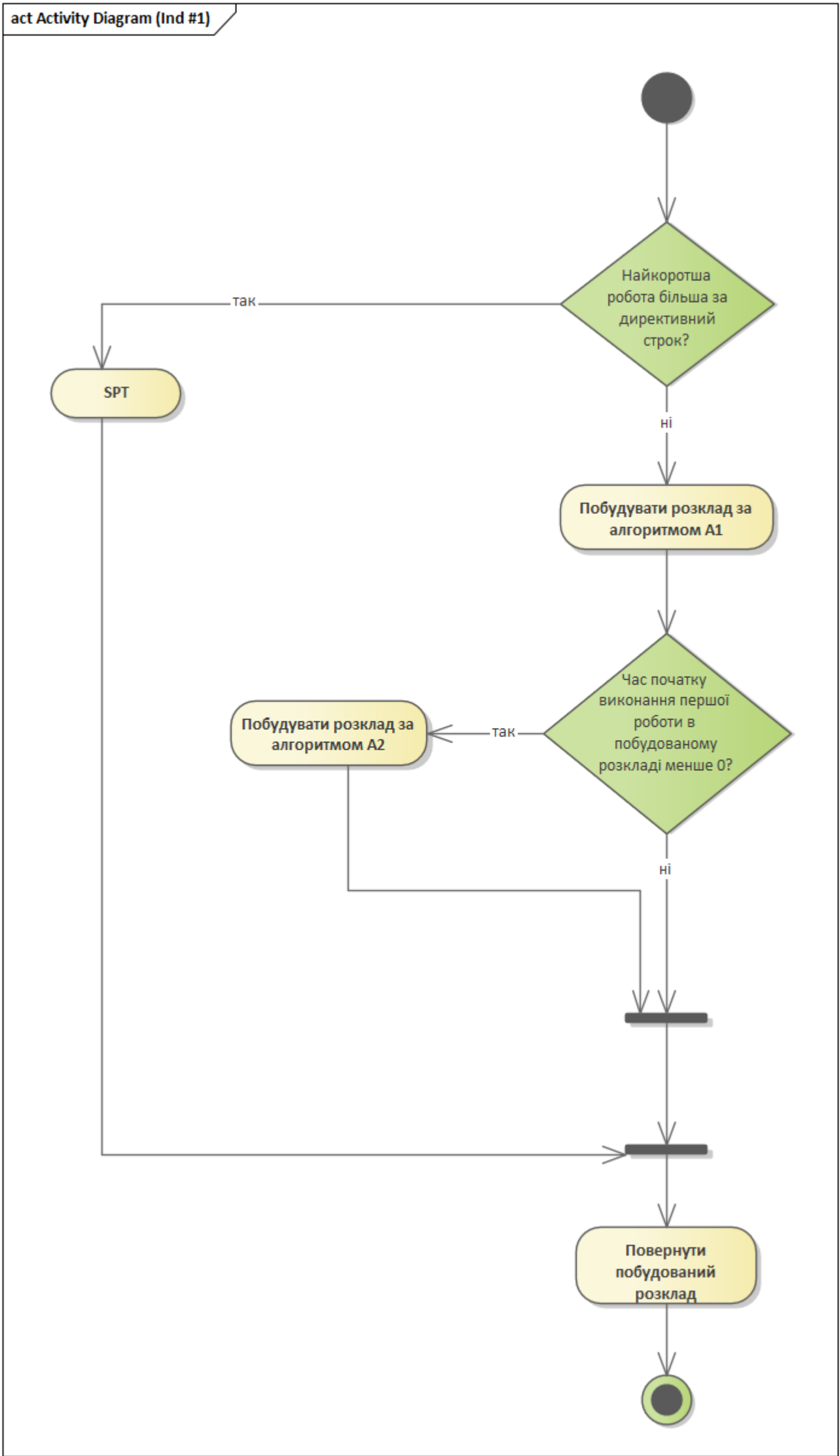
Підміна символів

Не знайдено заміненних символів

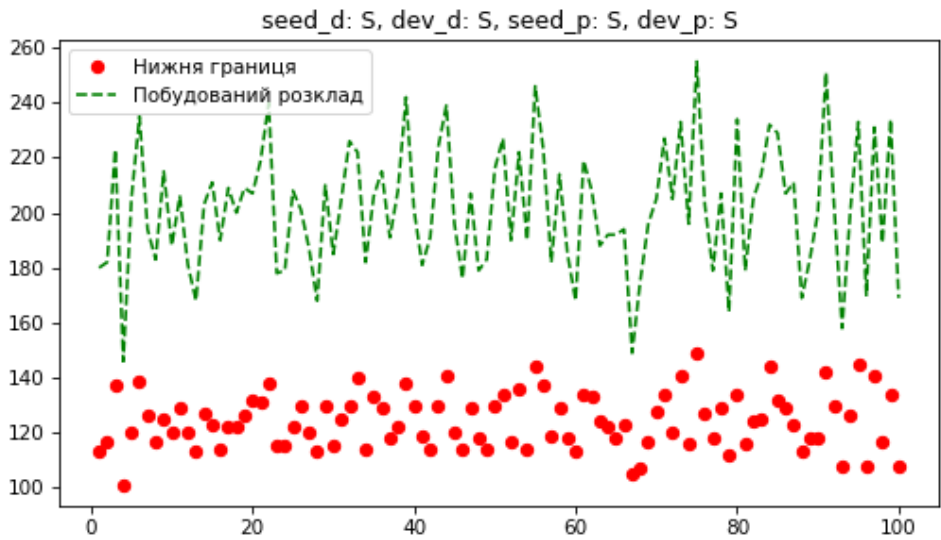
Графічний матеріал до дипломного проєкту

на тему: Система з підтримки процесу дослідження алгоритмів розв'язання
деякого класу задач теорії розкладів. Задачі з одним директивним строком.
Підсистеми візуалізації та аналізу результатів.

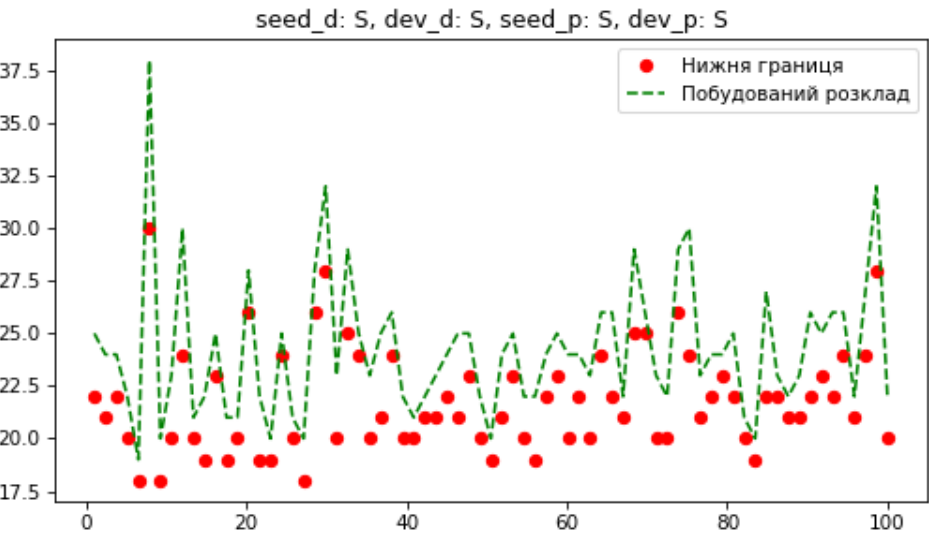
Київ – 2020 року



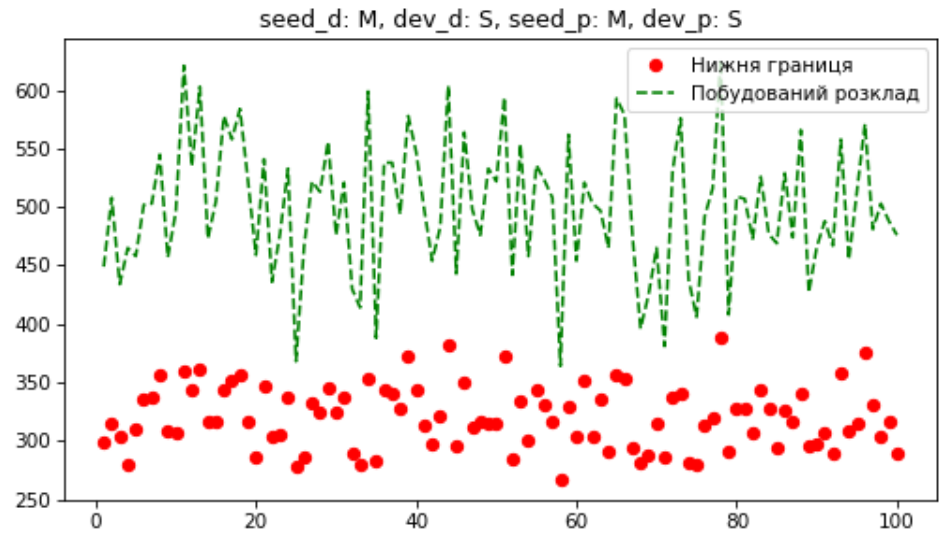
					ДП ІС-6119.06.000 ССД									
					Схема структурна діяльності	Лит.			Маса		Масштаб			
Зм.	Арк.	№ докум.	Підпис	Дата										
Розробн.		Омельченко І.О.												
Перев.		Жданова О.Г.				Аркуш 1			Аркушів 1					
Т. Кон.					Система з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач теорії розкладів. Задачі з одним директивним строком. Підсистеми візуалізації та аналізу результатів						КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-61			
Н. Контр.		Телишева Т.О.												
Затв.		Жданова О.Г.												



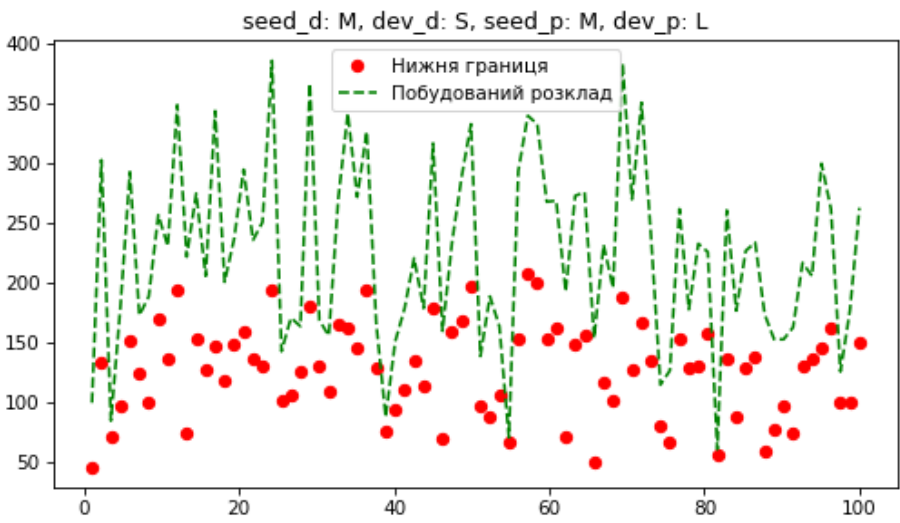
Результати дослідження при маленьких значеннях директивного строку та розкиду тривалостей робіт на декількох машинах



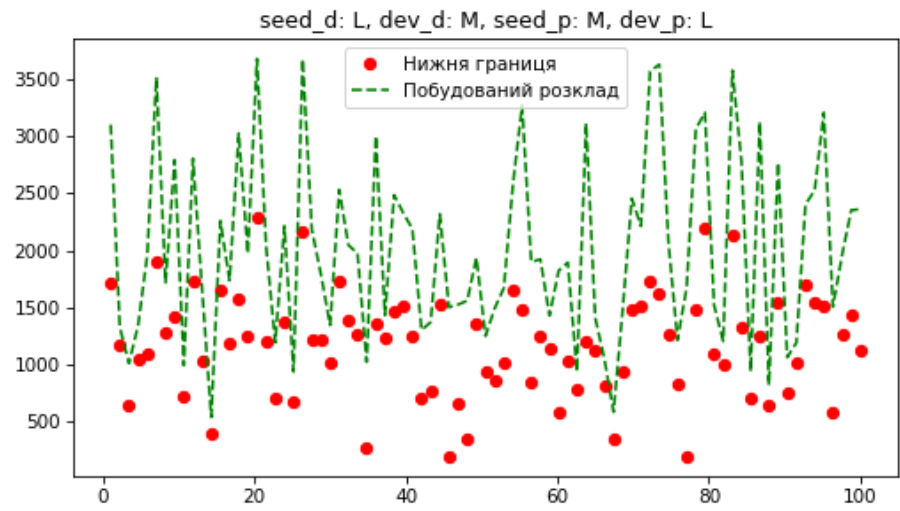
Результати дослідження при маленьких значеннях директивного строку та розкиду тривалостей робіт на одній машині



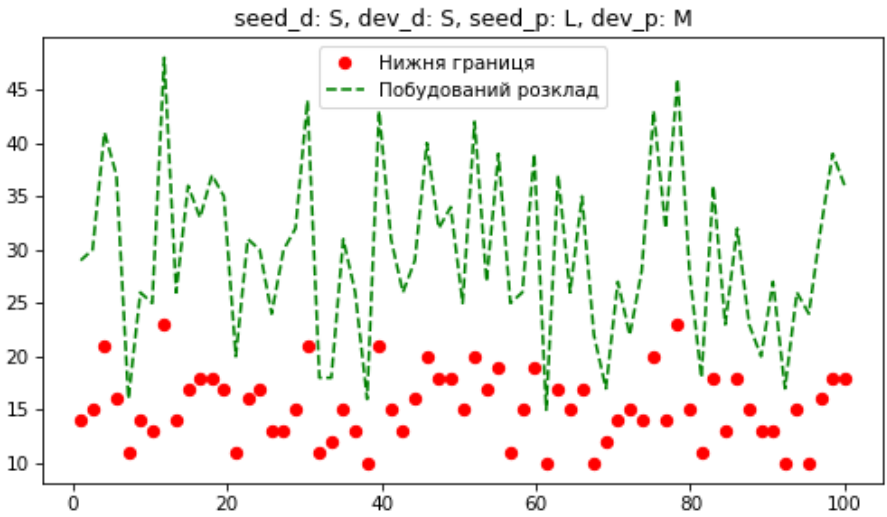
Результати дослідження при маленьких значеннях директивного строку та середніх значеннях розкиду тривалостей робіт на декількох машинах



Результати дослідження при маленьких значеннях директивного строку та середніх значеннях розкиду тривалостей робіт на одній машині



Результати дослідження при великих значеннях директивного строку та середніх значеннях розкиду тривалостей робіт на одній машині

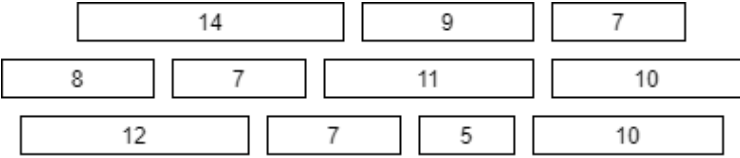


Результати дослідження при малих значеннях директивного строку та великих значеннях розкиду тривалостей робіт на одній машині

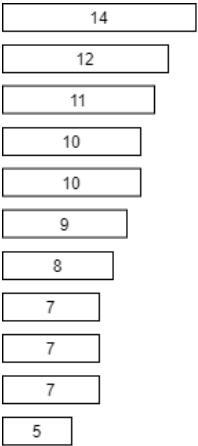
					ДП ІС-6119.07.000 КЗ							
						Креслення вигляду звітних форм	Літера			Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата								
Розробн.		Омельченко І.О.										
Перевірів		Жданова О.Г.				Аркуш 1			Аркушів 1			
Т. кон.					Система з підтримки процесу дослідження алгоритмів розв'язання деякого класу задач теорії розкладів. Задачі з одним директивним строком. Підсистеми візуалізації та аналізу результатів	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-61						
Н. контр.		Тєлишева Т.О.										
Затвердив		Жданова О.Г.										

Рішення з математичного забезпечення

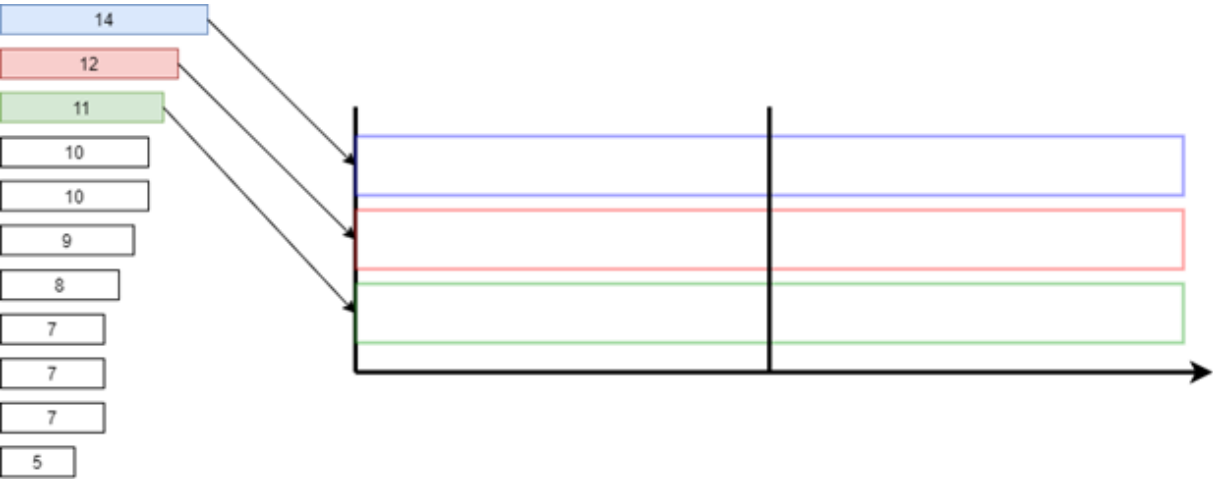
Дано множину робіт. Необхідно розподілити їх між трьома ідентичними машинами



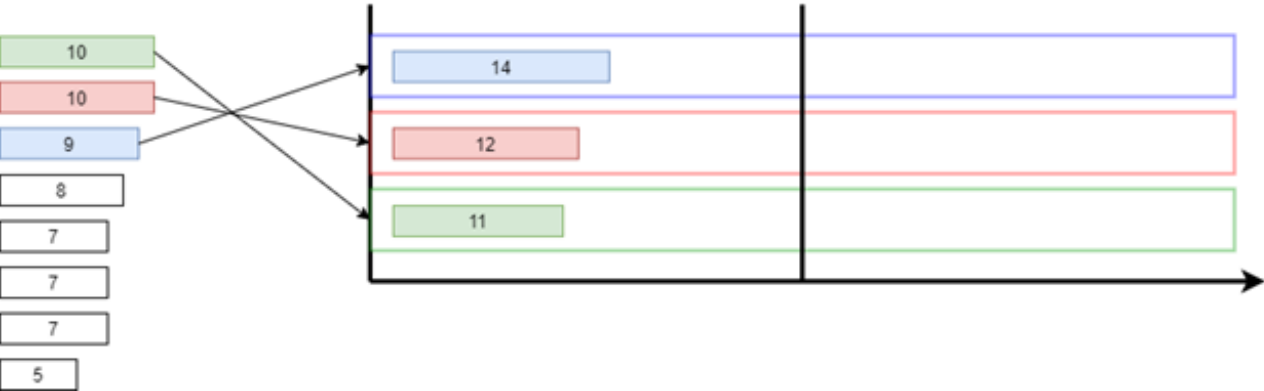
КРОК 1: відсортуємо роботу у порядку спадання тривалостей



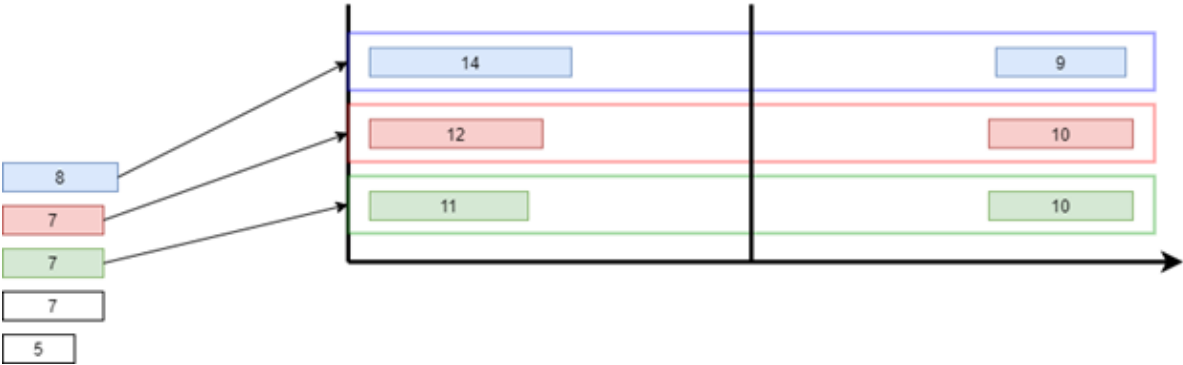
КРОК 2: перші m робіт (де m – кількість машин, в даному випадку $m=3$) розподіляємо в звичному порядку між машинами. Тобто перша робота йде до першої машини, друга до другої, третя – до третьої.



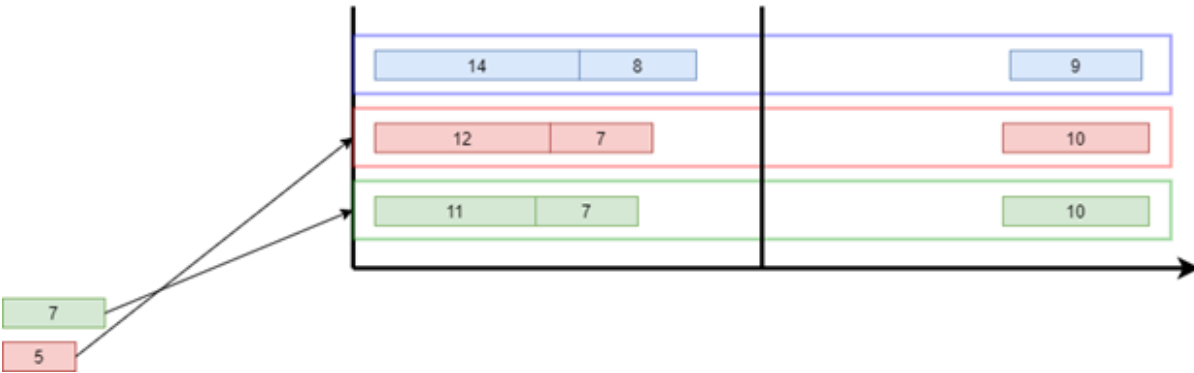
КРОК 3: наступні m робіт розподіляємо в протилежному порядку, тобто починаючи з останньої машини. Перша робота йде до третьої машини, друга до другої, третя – до першої.



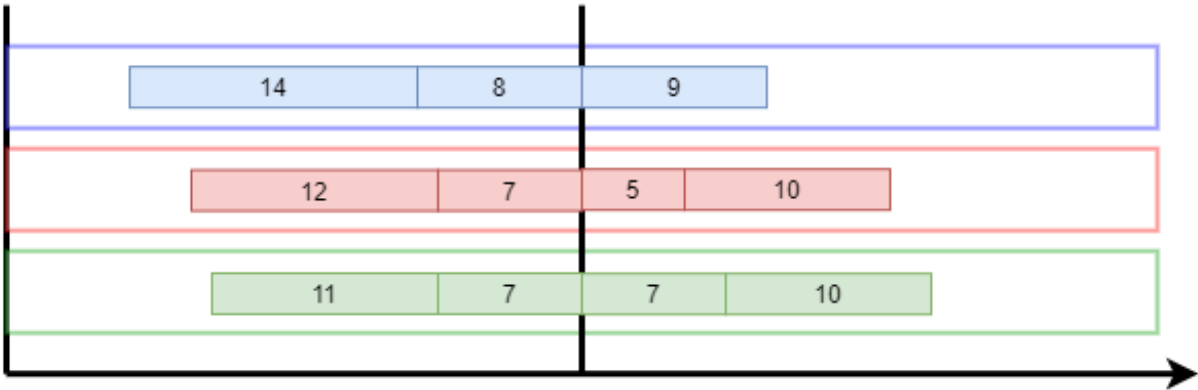
КРОК 4: наступні m робіт розподіляємо знову за звичним порядком



КРОК 5: останні роботи знову розподіляємо у зворотньому порядку



Отримали наступний розподіл



Демонстраційний плакат до дипломного проекту

„Система з підтримки процесу дослідження алгоритмів розв’язання деякого класу задач теорії розкладів. Задачі з одним директивним строком. Підсистеми візуалізації та аналізу результатів”

Виконала студентка гр. ІС-61

Омельченко І.О.

Керівник ДП

Жданова О.Г.