Національний університет «Полтавська політехніка імені Юрія Кондратюка»
(повне найменування вищого навчального закладу)

Навчально-науковий інститут інформаційних технологій та робототехніки
(повна назва інституту)

Кафедра комп'ютерних та інформаційних технологій і систем
(повна назва кафедри)

Пояснювальна записка
до дипломного проекту (роботи)
магістра
(рівень вищої освіти)
на тему
Розробка web-додатку на базі генетичного алгоритму для задач розкрою в умовах фіксованих 2-Dобмежень

Виконала: студентка2 курсу, групи601-ТН
спеціальності
122 Комп'ютерні науки
(шифр і назва спеціальності)
КлочкоА. О.
(прізвище та ініціали)
Керівник СкакалінаО. В.
(прізвище та ініціали)
Рецензент
(прізвище та ініціали)

Полтава – 2021року

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ « ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ

КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

спеціальність 122 «Комп'ютерні науки»

на тему

«Розробка web-додатку на базі генетичного алгоритму для задач розкрою в умовах фіксованих 2-D обмежень»

Студентки групи 601-ТН Клочко Анастасії Олександрівни

Керівник роботи
кандидат технічних наук,
доцент СкакалінаО.В.

Завідувач кафедри
кандидат технічних наук,
доцентГоловко Г. В.

Полтава – 2021

РЕФЕРАТ

Кваліфікаційна робота магістра: 118 с., 66 малюнки, 6 додатки, 89 джерел.

Об’єкт дослідження: генетичний алгоритм для задач розкрою в умовах фіксованих 2-Dобмежень.

Мета роботи: розробкаweb-додатку на базі генетичного алгоритму для задач розкрою в умовах фіксованих 2-Dобмежень.

Методи: проектування та створення web-додатку на базі генетичного алгоритму для задач розкрою в умовах фіксованих 2-Dобмежень, розробка макетів користувацького інтерфейсу.

Ключові слова:генетичний алгоритм, метод оптимізації, розкрій, 2-D розкрійзадача розкрою, особа, кросинговер, мутація, відбір, web-додаток.

ABSTRACT

Explanatory note contains: 118 pages, 66 pictures, 6additions, 89 references.

Object of research – genetic algorithm for cutting problems under conditions of fixed 2-D constraints.

The purpose of the qualification work: development of a web-application based on a genetic algorithm for cutting tasks under fixed 2-D constraints.

Methods: design and creation of a web-application based on a genetic algorithm for cutting tasks in conditions of fixed 2-D constraints, development of user interface layouts.

Keywords: genetic algorithm, optimization method, cutting, 2-D cutting, cutting problem, person, crossover, mutation, selection, web-application.

ЗМІСТ

ПЕРЕЛІКУМОВНИХПОЗНАЧЕНЬ, СИМВОЛІВ,СКОРОЧЕНЬ І ТЕРМІНІВ

ГА – генетичний алгоритм;

Індивід – (генетичний код) набір хромосом;

ІТ – інформаційні технології;

Кросовер – (перехресний перехід) це процес, при якому 2 хромосоми змінюють свої частини;

ЛДСП– ламінована деревостружкова плита;

МГА – мультиметодний генетичний алгоритм;

Мутація– це випадкова заміна одного або декількох частинок хромосоми;

ОБ – ортогональний багатогранник;

ОС – операційна система;

ПК – персональний комп’ютер;

Хромосома – це вектор (послідовність) нулів і одиниць;

1DBPP– клас завдань двовимірної упаковки на нескінченну смугу («1-One-DimensionalBinPackingProblem »);

2DBPP – клас задач двомірної контейнерної упаковки прямокутників («Two-DimensionalBinPackingProblem»);

3DBPP – клас завдань тривимірного контейнерного пакування паралелепіпедів («Three-DimensionalBinPackingProblem»);

BF – евристика «найкращий підходящий» («BestFit»);

BL – евристика «нижній лівий» («BottomLeft»);

BPP– клас завдань контейнерноїупаковки («BinPackingProblem»);

CHC – Cross generational elitist selection, Heterogenous recombination, Cataclysmic mutation;

CJM – картка подорожі клієнта(«Customer Journey Map»);

CSP – клас завдань розкрою («CuttingStockProblem»);

CSS –каскадні таблиці стилів («CascadingStyleSheets»);

FF – евристика «перший підходящий» («FirstFit»);

F-ShapedPattern – F-образна траєкторія перегляду веб-сторінки;

HTML – мова розмітки гіпертекстових документів(«HyperTextMarkupLanguage»);

HTTP – протокол передачі гіпертексту («HyperTextTransferProtocol»);

HTTPS – розширення протоколу HTTP для підтримки шифрування («HyperTextTransferProtocolSecure»);

HUX – однорідний кросовер («Half Uniform Crossover»);

IGES – цифрове уявлення обмінюватись даними визначальнимипродукту («Initial Graphics Exchange Specification»);

IIPP – клас завдань заповнення («Identical Item Packing Problem»);

JS – прототипно-орієнтована мова програмування(«JavaScript»);

KP – клас завдань про ранець («KnapsackProblem»);

NF – евристика «наступний» («NextFit»);

MPLP – завдання пакетування («Manufacturer's Pallet Loading Packing Problem»);

NP – клас складності задач з недетерміновано-поліноміальними алгоритмами («NondeterministicPolynomial»);

ODP – клас завдань упаковки у відкриту область («OpenDimensionProblem»);

PP – клас завдань розміщення («Placement Problem»);

QA – забезпечення якостіфункціонального тестування («QualityAssurance»);

SPP – клас завдань рулонного розкрою («StripPackingProblem»);

SSL – криптографічний протокол («Secure Sockets Layer»);

Statechart Diagram– діаграми станів;

STL – файл моделювання для стереолітографії(«StereoLithography»);

TLS–протокол захисту транспортного рівня («Transport Layer Security»);

TS– алгоритм пошуку з заборонами або табу-пошук («TabuSearch»);

UML–діаграма прецедентів («Unified Modeling Language»).

ВСТУП

Сьогодні про генетичні алгоритми можна говорити як про метод, яким вирішено безліч різних завдань. Про це свідчить величезна бібліографія, в тому числі і українською мовою. Проте, генетичні алгоритми не перестають бути предметом суперечок про ефективність їх роботи і доцільності їх використання.

Однією з головних завдань промисловості є зниження витрат виробництва, в тому числі і шляхом оптимального використання матеріалів і ресурсів. У багатьох галузях промисловості потрібен розкрій матеріалу.

Ефективне витрачання сировини – актуальна і складна задача, яка становить проблему промислового виробництва, при промисловому розкрої різних матеріалів типу листів металу, скла або дерева, труби, профільного прокату, вироби складної форми. Для вирішення такого завдання необхідно досягти максимально вигідного використання матеріалу, з якого вирізають заготовки, що по суті і є раціональним розкриємо матеріалу, тобто таким розкриємо, при якому досягається найменша кількість непотрібних відходів (залишків). Ефективне використання сировинного матеріалу прямо пропорційно впливає на економічну ефективність. Кількість відходів становить значний відсоток, помітно впливає на загальний бюджет підприємств, адже крім прямих витрат на дорогі матеріали існують і другорядні, наприклад, такі, як транспортування, складування та утилізація.

Метою даної роботи є розробка web-додатку на базі генетичного алгоритму для задач розкрою в умовах фіксованих 2-D обмежень, для компаній які займаються виробництвом корпусних меблів.

Для досягнення даної мети, потрібно розв’язати завдання:

– проаналізувати актуальні наукові дослідження в області рішення задач 2-D розкрою на основі генетичних алгоритмів;

– проаналізувати існуючі додатки аналоги для рішення задач розкрою в умовах фіксованих 2-D обмежень;

– спроектувати функціонал та структуру web-додатку, розробити макети екранів майбутнього додатку;

– спроектувати генетичний алгоритм для рішення задач плоского розкрою в умовах фіксованих 2-D обмежень;

– розробити, протестувати та ввести в експлуатацію власний програмний продукт на основі генетичного алгоритму в умовах фіксованих 2-D обмежень для компанії що займається виготовленням корпусних меблів.

Можливими галузями застосування результатів даної роботи є проектування та розробка корпусних меблів. Використання програмного забезпечення в рази полегшує проектування моделей, розкрій меблевих листів та виготовлення деталей під час виробництва меблів.

Як результат, розроблений додаток повинен надавати повний обсяг функцій для роботи майстрів що займаються виробництвом корпусних меблів, а саме:

– можливість додавання деталей з фіксованим розміром;

– можливість додавання нових листів матеріалу їх кількість та ціну за погонний метр;

– вивід діаграми оптимізованого розкрою матеріалу на основі генетичного алгоритму;

– прорахування сумарної площі листа матеріалу, деталей та відходу;

– прорахування ціни використаного матеріалу;

– автоматичне формування звіту про результати роботи над проектом;

– можливість перегляду короткої довідки щодо роботи з додатком.

РОЗДІЛ1
АНАЛІТИЧНИЙ ОГЛЯДWEB-ДОДАТКІВ НА БАЗІ ГЕНЕТИЧНИХ АЛГОРИТМІВ В УМОВАХ ФІКСОВАНИХ 2-DОБМЕЖЕНЬ

1.1 Описпредметноїобласті

Різноманітність завдань комбінаторної оптимізації, таких як упаковки та розкривання, а також суміжних завдань маршрутизації, складання розкладів та іншого, є проблемою як теоретичного, так і практичного плану. Причина зростаючого інтересу до цих завдань з боку виробничників полягає у їхньому безумовно великому прикладному значенні, а з боку дослідників у різноманітності та складності, пов'язаної з пошуком досить точного рішення у просторі великої розмірності.

Вхідними даними для розв’язування задачі розміщення довільних об’єктів на площині ємасив лекал, які необхідно розмістити, і масив контурів, в яких потрібно розташувати лекала. Саміконтури можуть бути як прямокутної, так і довільної форми, тобто матеріал міг використовуватисяв попередніх задачах і залишився у відходах. Так само і викрійки – можуть мати правильну інеправильну форму. Насамперед варто визначити, в якому контурі можна розташувати вибрані з масиву деталі. У випадку неможливості розміщення жодної деталів контурі, вінвважається відходним матеріалом, і видаляється з масиву контурів.

Для роботи генетичного алгоритму насамперед нам потрібно визначити, за якими критеріями необхідно вибирати лекала і контури, в які вони будуть розміщуватися. Оскільки напередневідомо, якої форми в нас будуть лекала і контури площин, на які ми будем розміщувати лекала,необхідно розглядати різні критерії. Спочатку треба розглядати критерії, які будуть опиратися наплощу і форму лекал, хоча не треба відкидати і інші можливі критерії.

Визначення таких критеріїв дасть змогу зробити рейтингування критеріїв за ефективністю їхвикористання з подальшим врахуваннях цієї ефективності у

експертній підсистемі.

Для розташування лекал за контурами на основі генетичних алгоритмів було розробленоалгоритм розташування лекала за контуром довільної форми. Було розглянуто такі критерії виборулекал для розміщення їх на площинах:

– максимальна площа;
– мінімальна площа;
– мінімальний периметр;
– максимальний периметр;
– мінімальна описана площа прямокутника;
– максимальна описана площа прямокутника.

Цей список не претендує на повноту, але ці критерії повинні бути базовими для розв’язаннязадач розкрою за допомогою генетичних алгоритмів.

Наприклад, такий критерій, як максимальна площа лекала дасть змогу відразу розміститинайбільше лекало і зменшить можливість того, що це лекало не буде розміщене. А мінімальнаплоща лекала дасть можливість розмістити найменші лекала в таких місцях контуру, де великі непомістяться, що дасть змогу зменшити відходи.

Залежно від різновидів задач розкрою, розробники можуть додавати інші критерії, які, наїхню думку, можуть істотно вплинути на результат виконання поставленої задачі.

Ці критерії також підходять і для вибору контурів. Для роботи алгоритму необхідно ввестикритерії для розміщення лекала в контурі, а саме:

– кількість спільних сторін лекала і площини (цей критерій дає можливість розміститилекало впритул до контуру площини і тим самим зменшити можливі втрати матеріалу);
– кількість утворених контурів після розміщення лекала;
– кількість непридатних контурів, які після розміщення лекала будуть відкинуті (цейкритерій дає можливість вибрати таке розміщення, за якого утвориться найменша кількістьконтурів).

До цих розглянутих критеріїв можуть додатися і інші, які будуть давати позитивнийрезультат для певного різновиду задач розкрою.

Для роботи генетичного алгоритму, створюється початкова популяція. Популяція – це набіргенів. У нашому випадку ген – це набір критеріїв для розміщення лекал на площині.

Наприклад: (a,b,c) – це ген,

де a – критерій вибору лекала;

b – критерій вибору контуру, на який буде розміщене лекало;

c – критерій вибору розміщення лекал.

Початкова популяція може створюватись як зі всіх можливих генів, так і мати певніобмеження. Тобто, є можливість існування тільки певної кількості генів. Після утворення початкового покоління кожен ген оцінюється на можливість продовження роду, тобто обчислюється функція корисності. У нашому випадку ми оцінюємо, за допомогою якого гена в задачі розкрою буде отримана найменша площа відходів. Оцінивши результати –відбираємо приблизно 30% генів, які показали найкращий результат. Цей процес називається селекцією. Із збережених генів (проміжне покоління) за допомогою генетичних операцій – схрещення і/або мутації – ми утворюємо нове покоління і оцінюємо його спроможність щодо подальшого існування.

Для задач розкрою пошук вдосконалих генів продовжується доти, доки не розмістяться всілекала або не закінчуються контури (рисунок1.1).

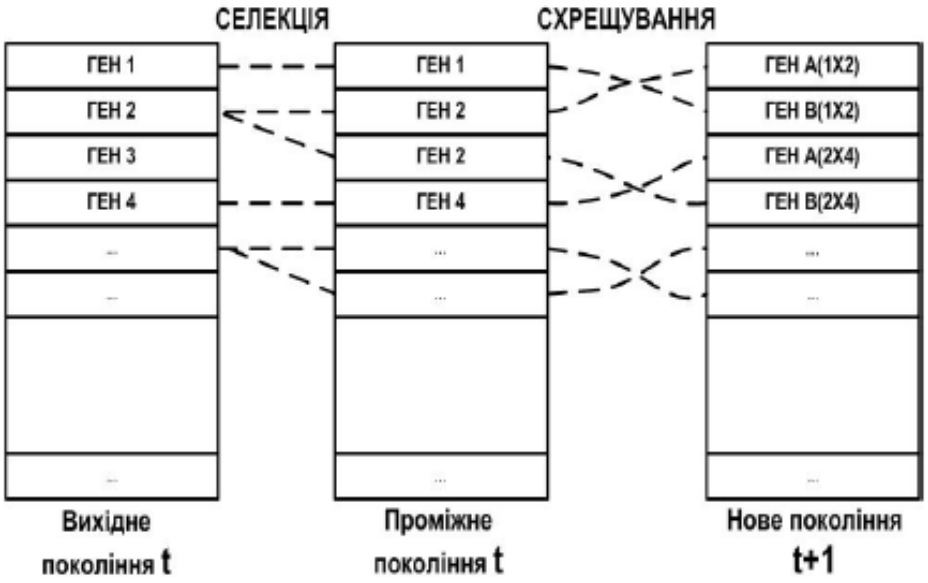


Рисунок 1.1 – Схема пошуку генів для нової популяції

Вхідними даними для роботи розробленого алгоритму розташування лекал довільної формина площині є масив лекал, які необхідно розмістити, і масив контурів, в яких потрібно розташуватилекала.

1.2 Огляд наукових досліджень в області рішення задач 2-Dрозкрою

Для проведення наукових досліджень студент повинен розвивати гарні навички в зборі та обробці різних матеріалів, аналізувати і критично оцінювати теоретичні положення та існуючі методи у відповідній сфері діяльності. Перша важка частина дослідження – виявити проблему, вирішення якої дійсно принесе користь суспільству і успіху здобувача. Визначення проблеми здійснюється на основі критичного аналізу спеціальних літературних джерел і стану справ на практиці.

1.2.1 Огляд основних видів генетичних алгоритмів. Генетичний метод є спільною моделлю еволюції в природі, яка застосовується під виглядом комп'ютерногододатку. Використовується як аналогприродного відбору. Однак біо-термінологія злегка зберігається [1].

Моделювання генетичного спадкування:

– хромосома – це вектор (послідовність) нулів і одиниць.Будь-яка позиція називається геномом;

– індивід (генетичний код) являє собою набір хромосом (можливість виконання завдання);

– кросовер (перехресний перехід) – це процес, при якому 2 хромосоми змінюють свої частини;

– мутація – це випадкова заміна одного або декількох частинок хромосоми.

Для створення еволюційного процесу, ми спершу створюємо випадковий набір популяції – певну кількість особин з довільним набором хромосом . ГА імітує еволюцію даної популяції як повторюваний процес схрещування особин і плину поколінь (рисунок1.2).

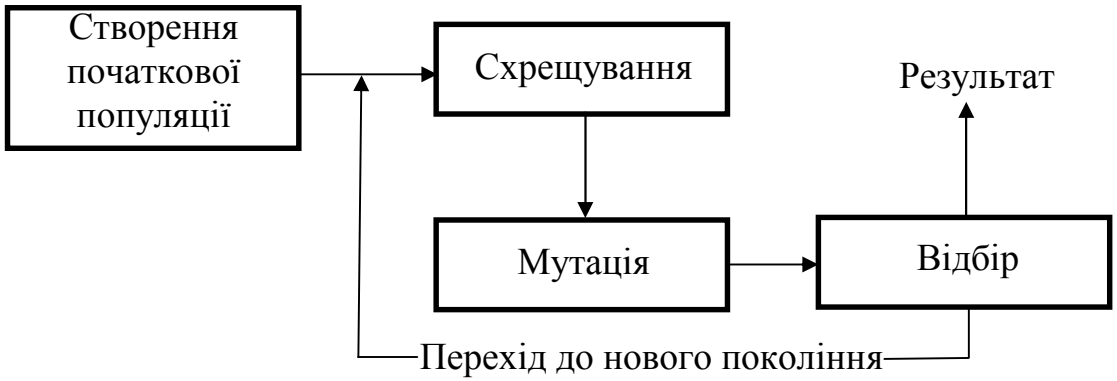


Рисунок1.2 – Загальна схема роботи генетичного алгоритму.

Життєпис популяції – це деяка кількість непередбачених схрещувань (у вигляді кросовера) і мутацій, після яких додається кілька нових індивідуумів в популяцію. Відбір у ГА – це формування зі старої популяції нову, далі йде знищення старої. Після відбору в нову популяцію знову використовуються операції мутації і кросовера , потім знову відбувається відбір, і так далі.

Відбір у ГА тісно пов'язаний з принципами природного відбору:

– придатність індивіда – це важливість цільової функції над цим

індивідом;

– виживання більш адаптованих – нове покоління населення формується у гармонії з цільовою функцією;

– чим більш пристосований індивід, тим більша ймовірність участі у кросовері, тобто тим більша спадковість.

Отже, модель відбору передбачає, як формуватиметься нова генерація населення. Зазвичай, можливість участі індивіда у гібридизації приймається поки він придатний.Стратегію *елітарності* використовують, коли найкращі індивіди переходять до нової генерації незмінними, не зважаючи на участь у переходах та виборах. Кожна нова генерація буде зазвичай кращою за попередню. Коли придатність індивідів перериває процес покращення, рух припиняється, і вони залишають найкращих з тих, хто знаходить вирішення проблем оптимізації.

Слід зазначити, що в даний час ГА є цілим класом алгоритмів, спрямованих на вирішення різних проблем.

Існують різні моделі ГА (класичний, традиційний ГА, гібридний, ГА CHC та ін.). Вони відрізняються по стратегії відбору та формування нового покоління особин, оператори генетичних алгоритмів, кодування генів і так далі.

CHC-алгоритмбув запропонований Есхелманом і характеризується наступними параметрами:

– для нового покоління вибираються N найкращі декади між батьками та дітьми. Дублювання рядків не допускається;

– для декомпозиції вибирається випадкова пара, але не допускається надто мала відстань декомпозиції між батьками або надто невелика мала між надмірно різними бітами;

– для переходу використовується якийсь однорідний кросовер HUX: через дитину проходить рівно половина бітів кожного з батьків;

– чисельність популяції невелика, близько 50 особин. Це виправдовує використання однорідного переходу.

CHC суперечить агресивному відбору проти агресивного переходу, але все ж таки невеликий розмір популяції швидко призводить до ситуації, коли формуються більш-менш однакові індивіди. В даному випадку CHC використовує cataclysmicmutation: ряди, крім найбільш пристосованих, піддаються сильній мутації (змінюється приблизно третя частина бітів). Отже, алгоритм перезапускається та продовжує працювати, застосовуючи лише один прохід[2].

Genitorбув створений Д. Уітлі. Genitor-подібні методи відрізняються від традиційного ГА належними трьома якостями:

– на кожному кроці тільки 1 пара випадкових опікунів робить тільки 1-го нащадка;

– цей нащадок заміщає не батька, а одну з гірших особин популяції (в початковому Genitor – найгіршу);

– відбір особини для заміни здійснюється за його рейтингом, а не за пристосованістю.

У Genitor розвідка гіперплощин буває кращою, а збіжність швидше, ніж у традиційного генетичного методу, запропонованого Холландом[3].

Ідея **гібридних алгоритмів** полягає у поєднанні генетичного способу з деяким іншим методом пошуку, відповідно даної задачі. У кожному поколінні кожен отриманий нащадок оптимізується обраним методом, після чого виконуютьсяпритаманні для ГАдії.

Подібна картина становлення називається ламарковою еволюцією, при якій індивід може навчатися, а потім отримані можливості записувати в індивідуальний генотип, щоб потім передати їх нащадкам. Такий спосіб послаблює можливість способу шукати висновки за допомогою відбору гіперплощин, однак на практиці гібридні способи виявляються досить вдалими. Це пов'язано з тим, що зазвичай велика ймовірність того, що одна з особин потрапить до району масового максимуму і після оптимізації виявиться рішенням задачі[4].

Переважна більшість завдань оптимізації вимагають величезних витрат

часу та обчислювальних ресурсів. Це пов'язано з необхідністю перебирати величезну кількість різноманітних рішень. Більше того, за своєю обчислювальною складністю такі задачі належать до класу так званих NP-повних задач, тобто задач, для яких не існує детермінованого поліноміального алгоритму[5]. Тому для того, щоб у таких задачах було знайдено найкраще рішення (глобальний екстремум), необхідно здійснити вичерпний пошук, який насправді неможливий через їх велику розмірність [6, 7]. Тому на практиці для вирішення таких задач розробляються різноманітні метаевристичні алгоритми, які дозволяють знаходити близькі до оптимальних (квазіоптимальні) рішення. Одним із підходів, що дозволяє успішно вирішити проблему підвищення ефективності та якості вирішення складних оптимізаційних задач великої розмірності, є інтеграція різноманітних наукових методів, специфічних для таких галузей обчислювального інтелекту, як біоінспіровані алгоритми, нечіткі обчислення, штучні нейронні мережі[8]. Генетичні алгоритми (ГА) як представники групи метаевристик стали важливим інструментом для вирішення оптимізаційних задач у різних галузях [5].

Паралельні генетичні алгоритмиможливо виконати як кілька одночасно запущених процесів, це підвищить їх результативність.

Розглянемо перехід від традиційного ГА до паралельного. Для цього застосуємо турнірний відбір. Припустимо, що маємо N/2 виробок, кожна з них буде обирати довільно з популяції 4 особини, проводиться 2 турніри після чого лідерівсхрещують. Одержані нащадки будуть записані в нову генерацію. Як результат, прогін роботи одного процесу відразу буде змінювати ціле покоління[9].

Острівна модельтакож є паралельною моделлю ГА. Її суть: припустимо ми маємо 16 процесів та 1600 осіб. Розберемо їх на 16 субпопуляцій по 100 індивідуумів. Кожен розвиватиметься за допомогою обраного ГА окремо. Отже, можна сказати, що ми розмістили даних особин на 16-ти ізольованих окремих островах (рисунок1.3).

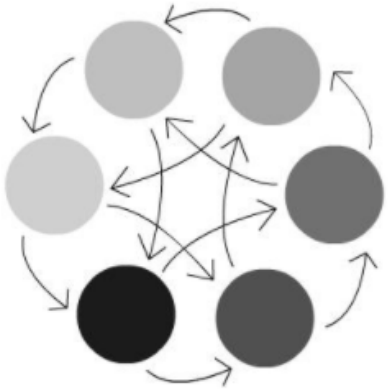


Рисунок1.3 – Острівна модель генетичного алгоритму

Інодіпроцеси (острови) будуть перекидатися декількома вдалими особинами. Цей процес називається міграцією. Дане переміщення дозволяє островам перекидатися генетичним матеріалом.

Наприклад, коли заселеність острівців мала, підпопуляції будуть близькі до дострокової ідентичності. Тому значущу роль має коректно прорахована періодичність пересування. Дуже часте переміщення (чи пересування дуже великої кількості осіб) як наслідок призведе з часом до змішування підпопуляцій, за цей час острівна модель стане не сильно різнитися з простим ГА. Якщо ж переміщення стане дуже рідким, то воно не може попередити раннє сходження підпопуляцій.

ГА стохастичні, тому при різних прогонах населення може зближуватися з різними наслідками (хоча всі вони певною мірою «хороші»). Острівна модель дозволяє запустити спосіб одночасно кілька разів і намагатися поєднувати «досягнення» різних островів для отримання в одній з підпопуляцій кращого рішення[10].

Ніздрюваті ГА–модель паралельних генетичних алгоритмів. Припустимо дано 2500 процесів, що знаходяться на сітці об'ємом 50×50 осередків, закритої, як показано на рисунку 1.4 (ліва сторона замикається з правою, нижня – з верхньої).

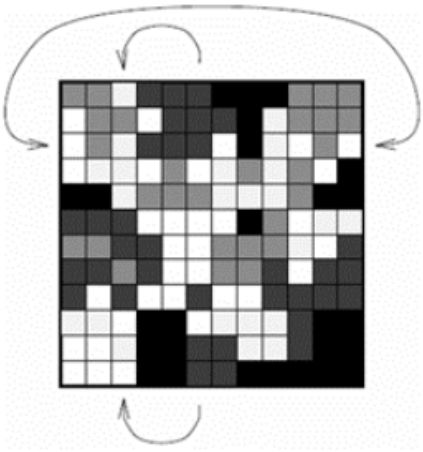


Рисунок1.4 – Ніздрюватий генетичний алгоритм

Будь-який процес може взаємодіяти лише з 4 власними сусідами (згори, знизу, ліворуч, праворуч). У кожному осередку розташовується рівно один індивід. Кожен процес буде обирати кращу індивідуум між своїх сусідів, схрещувати з нею індивід з власного осередку і одного отриманого малюка поміщати в осередок замість засновника.

При застосуванні такого способу виникають ефекти, аналогічні на острівну модель. Спочатку всі особини мають випадкову пристосованість (на малюнку вона орієнтується за кольором). Крізь кілька поколінь утворюються дрібні області схожих особин з найближчою пристосованістю. У міру роботи ГА ці області виростають і конкурують одне з одним[11].

1.2.2 Аналіз предметних областей для формування задачі генетичних алгоритмів.Генетичні алгоритми – це адаптивні методи пошуку, які деконструюють рішення проблем функціональної оптимізації. Своєрідні моделі машинного дослідження пошукового простору, ґрунтуються на еволюційній метафорі і являють собою моделі машинного дослідження. Характерні особливості:

– використання термінів фіксованої довжини для подання інформації в генах;

– робота з населенням рядів;

– використання генетичних операторів для формування майбутніх поколінь.

Генетичні алгоритми працюють з групою осіб (популяцією), які представлені векторами, що кодують одне з рішень проблеми [12]. Цей метод відрізняється від більшості інших алгоритмів оптимізації, які працюють тільки з одним рішенням і удосконалюють його.

ГА використовують для розв’язку наступних проблем[13]:

– оптимізація функцій;

– дискретна оптимізація [14];

– різні завдання на графіку (завдання комівояжера, книжка-розмальовка і т. д.);

– налагодження та навчання штучної нейронної мережі;

– завдання компонування;

– створення розкладу;

– вантажоперевезення повітряним, морським, залізничним і автомобільним транспортом;

– стратегії гри;

– оптимізація вантаження [15];

– розподіл оперативної пам'яті і обчислювальних ресурсів в мультипроцесорних системах;

– наближення функцій;

– штучний інтелект;

– проектування компоновок простору літальних апаратів і схем розміщення вантажів на борту транспортних космічних систем;

– біоінформатики;

– рішення задач об’ємно-календарного планування;

– проектування компоновок теплоенергетичних схем і інженерних мереж;

– оптимізація запитів в базах даних;

– оптимізація управління проектами [16];
– синтез конструкцій антен;
– проектування компоновок надвеликих інтегральних схем;
– задачі логістики;
– прокладання маршруту [17].

Застосування ГА в області логістики було представлено в тезах до V All-Ukrainian scientific and technical conference "Computer Mathematics in Science, Engineering and Education CMSEE-2020" [18] та XIII All-Ukrainian scientific and practical conference "Actual problems of computer science APKN-2021" October 15-16, 2021 [19].

З матеріалами тез можна ознайомитись у Додатку Г.

Переваги генетичних алгоритмів:

– універсальність;
– висока видимість пошуку;
– в цільової функції немає обмежень;
– будь-який спосіб настройки функції.

Недоліки генетичних алгоритмів:

– відносно висока обчислювальна вартість;
– квазіоптимальність.

Коли потрібно використовувати генетичний алгоритм: безліч параметрів канави, слабкі цільові функції, комбінаторні проблеми.

Коли не слід використовувати генетичний алгоритм: проблема добре вирішується класичними методами, необхідна висока точність рішення.

1.2.3 Класифікація задач розкрою. Залежно від розмірності задачі, числа використовуваних контейнерів, асортименту розміщуваних об'єктів і обраної стратегії використання об'єктів і контейнерів, розрізняють велике число типів завдань розкрою-упаковки [20].

Вперше класифікація задач розкрою-упаковки була виконана

Л.В. Канторовичем і В.А. Заглаллером в 1971 р, проте ця класифікація булапідготовлена, в першу чергу, на основі публікацій вітчизнянихдослідників [21].

Перша детальна класифікація різних типів завдань розкрою-упаковки,прийнята у світовій практиці, була запропонована H. Dyckhoff в 1990 р [22]. Найбільшповний огляд наукових публікацій, підготовлених відповідно до цієїкласифікацією, виконаний H. Dyckhoff, J. Terno і G. Scheithauer [23].

В даний час серед дослідників в області оптимізації рішення задачрозкрою-упаковки визнана і найбільш широко використовується вдосконалена ідоповнена класифікація, запропонована G. Wäscher, H. Haubner і H. Schumann в2007 [24], при підготовці якої були розглянуті 445 наукових публікацій.

Первісна класифікація задач 2-Dрозкрою

Класифікація задач розкрою-упаковки, запропонована H. Dyckhoff [22], заснована на виділенні чотирьох ознак завдань, що характеризують об'єкти і контейнери, а також умови їх використання при формуванні упаковки. Представлені ознаки впорядковані за спаданням їх значимості при вирішенніпрактичних завдань розкрою-упаковки різних типів:

1. Розмірність (використовується в якості першої ознаки, оскільки обчислювальна складність рішення задачі, в першу чергу, визначається її розмірністю)

(1) –одномірні завдання (клас одновимірних задач є найбільшпроробленим з точки зору отримання оптимальних рішень);

(2) –двомірні задачі (цей клас задач має найбільше число областей практичного застосування, що пояснюється різноманітністю геометричних форм розміщуються об'єктів);

(3) – тривимірні задачі (для більшості завдань тривимірної упаковки розглядаються об'єкти в формі паралелепіпедів);

(N) – багатовимірні завдання (в якості додаткових розмірностей, якправило, використовуються непросторові величини, наприклад,час або

вартість).

2. Характер використання об'єктів і контейнерів:

(B) – всі задані об'єкти розміщуються в мінімальній кількості контейнерів (ця ознака використовується для позначення завдання мінімізації використовуваного простору контейнерів);

(V) – заповнення всіх заданих контейнерів об'єктами певних типів(ця ознака використовується для позначення завдання максимізаціївикористання вільного простору контейнерів).

3. Набір використовуваних контейнерів:

(O) – використовується один великий контейнер (ця ознака, зокрема,використовується для позначення завдання пакетування і завдання про ранці);

(I) – використовується набір ідентичних контейнерів (ця ознака, зокрема,використовується для позначення завдання контейнерної упаковкиоб'єктів);

(D) – використовується набір контейнерів, які мають різні геометричніхарактеристики (ця ознака, зокрема, використовується дляпозначення завдання розкрою матеріалу).

4. Набір розміщуваних об'єктів:

(F) – розміщення невеликого числа об'єктів (близько 10) різних типів(Наприклад, завдання завантаження багажника автомобіля);

(M) –розміщення великої кількості об'єктів різних типів (як правило, використовуються ідентичні контейнери);

(R) –розміщення великої кількості слабо розрізняються об'єктів (наприклад, задача розкрою листового матеріалу на тисячі об'єктів 50 типів);

(C) –розміщення ідентичних об'єктів (наприклад, завдання завантаження палетів);

На основі представлених ознак виконується кодування типів завданьрозкрою-упаковки, при цьому в найменуванні типу ознаки, розділені косою рискою,вказуються послідовно один за одним. наприклад:

– одномірна задача контейнерної упаковки (зокрема, завдання розподілу пам'яті в обчислювальній системі) позначається через 1/V/I/M;

– двомірна задача палетування позначається 2/B/O/C;

– завдання тривимірної контейнерної упаковки позначається через 3/V/ I або 3/ B/Ов залежності від заданих умов завдання;

– завдання багатоперіодну бюджетування позначається через N/B/O.

Сучасна класифікаціязадач 2-Dрозкрою

Первісна класифікація задач розкрою-упаковки (H. Dyckhoff) дозволиласистематизувати праці різних дослідників в області оптимізації рішеннязадач розкрою-упаковки, проте в подальшому вона потребувала подальшого доопрацювання, що пояснюється як появою нових типів завдань, так і появою новихтипів обмежень. Покращена актуальнакласифікація задач розкрою-упаковки, розроблена G. Wäscher, H. Haubner iH. Schumann, включає шість базових класів задач (рисунок 1.5) [24].

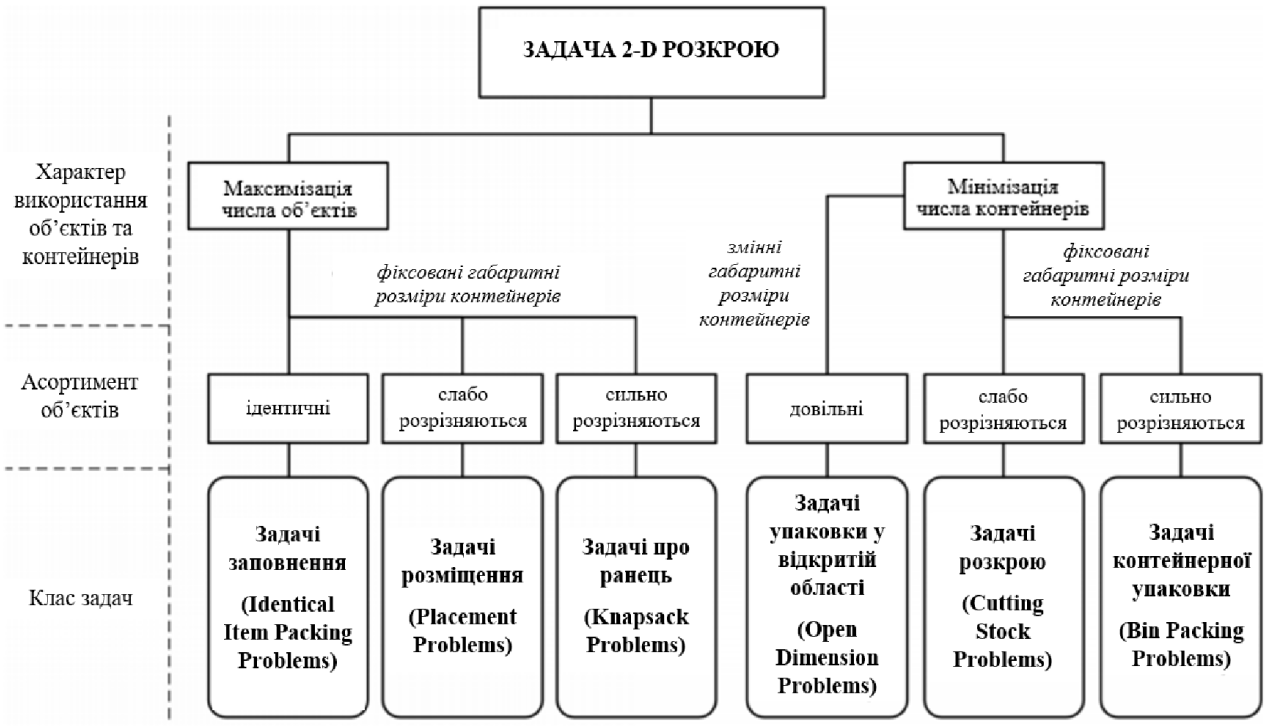


Рисунок 1.5 – Актуальна класифікація задач розкрою-упаковки (G. Wäscher, H. Haußner і H. Schumann)

I. Завдання заповнення («Identical Item Packing Problem», IIPP)

Ці завдання полягають в пошуку способу розміщення найбільшого числаідентичних об'єктів для заповнення всього простору контейнера. Оскількирозміщуються об'єкти можуть мати специфічну геометричну форму, для нихвідсутні універсальні алгоритми групування і кластеризації,забезпечують можливість розміщення об'єктів групами. Рішення задачізаповнення полягає в пошуку способу групування об'єктів для побудовигеометричного візерунка, що мінімізує загальний невикористаний обсягпростору контейнера.

Завдання заповнення в літературі також відома під назвою «завданняпакетування»(«Manufacturer's Pallet Loading Packing Problem», MPLP) [25].Прикладами завдань цього класу служать завдання заповнення поверхні однаковимиколами або еліпсами, а також завданнязаповнення обсягу контейнера кулями, циліндрами абоеліпсами.

Завдання заповнення найбільш часто зустрічаються в наступних областях:

– оптимізація зберігання запасів в складських системах (завдання заповненняпрямокутниками);

– проектування геометричної схеми захоронення циліндричнихконтейнерів з токсичними і радіоактивними відходами (завдання заповненняколами);

– проектування широкосмугових антен (завдання заповненняколами);

– планування процедури лазерної абляції при променевої терапії (завданнязаповнення колами);

– планування вантажоперевезень морським і залізничним транспортом(Завдання заповнення циліндрами);

– проектування фазованих антенних решіток (завдання заповненняплоскими елементами у формі Полімін);

– моделювання зернистих середовищ і проектування геометричнихконфігурацій молекул (завдання заповнення кулями або еліпсоїда);

– розкрій матеріалу в текстильній і взуттєвої промисловості (завданнязаповнення однаковими полігонами).

II. Завдання розміщення («Placement Problem», PP)

Завдання цього класу вирішують проблему пошуку способу розміщення найбільшогочисла об'єктів, слабо відрізняються один від одного габаритними розмірами, приумові мінімізації невикористаних вільних просторів використовуванихконтейнерів. На відміну від завдань класу IIPP, для задач класу PP потрібно завданнядекількох різних типів розміщуваних об'єктів [26, 27].

Завдання розміщення найбільш часто зустрічаються в наступних областях:

– розкрій листового матеріалу (завдання розміщення прямокутників);

– оптимізація зберігання складських запасів з використанням палет (завданнярозміщення прямокутників та паралелепіпедів);

– заповнення палуби судна (завдання покриття прямокутниками контейнера вформі багатозв'язного ортогонального полігону);

– тріангуляція геометричних фігур для побудови кінцево-елементнихсіток (завдання розміщення прямокутників, кіл і еліпсів різнихрозмірів);

– планування радіохірургічного лікування (завдання розміщення набору сферрізного радіусу);

– швидке прототипування при використанні технології селективноголазерного спікання (завдання розміщення багатогранників вциліндричному і сферичному контейнерах).

III. Завдання про ранці («KnapsackProblem», KP)

Для задач про ранці (або задач про рюкзак) розміщуються об'єкти за

габаритнимирозмірами повинні сильно відрізнятися один від одного.

У традиційній постановці завдання про ранці має такий вигляд. Єконтейнер вантажопідйомністю B> 0, званий «рюкзаком», а також набір з nоб'єктів з цінністюс$_i > 0$ і вагою b_i> 0 (i = 1, 2, . . . , n). Передбачається, що вагакожного об'єкта не перевищує значення B, при цьому сумарна вага всіх об'єктів більше вантажопідйомності контейнера.

У традиційній постановці завдання про ранці має такий вигляд. Дано набір з n об'єктів, кожен з яких має цінність ci> 0 і вага bi> 0, i = 1, 2, K, n, а такождано контейнер (так званий «рюкзак»), вантажопідйомність якого дорівнює B>0.Передбачається, що вага кожного об'єкта не перевищує значення B, але сумарна вагавсіх об'єктів більше вантажопідйомності контейнера. Необхідно визначити розташуванняоб'єктів всередині контейнера з максимальною цінністю без перевантаження (тобто цільовафункція має вигляд $\sum_{i=1}^{n} c_i\, x_i$ →optза умови $\sum_{i=1}^{n} b_i\, x_i \le R$, де $x_i \in \{0,1\} \forall\, i \in \{1,\ldots,n\}$)

Найбільш часто це завдання розглядається в одновимірній постановці [28,29], коли заданий одновимірний контейнер заданої довжини і набір відрізків різноїдовжини і різної цінності. Оптимальним вирішенням цього завдання буде набір відрізків максимальної цінності, сумарна довжина яких дорівнює довжині контейнера. Одновимірна задача про ранці досить добре вивчена, для цьогозавдання отримані апроксимаційні алгоритми вирішення поліноміальноїскладності [30].

Завдання про ранці має безліч додатків для двомірного, тривимірного ібагатовимірного випадків. У двомірному випадку рішенням завдання є розміщеннянабору прямокутників, що мають максимальну цінність. В тривимірному випадку оптимізується набір паралелепіпедів, що розміщуються вконтейнері, за критерієм максимізації цінності отриманої упаковки.

Розглянуті класи завдань (IIPP, PP і KP) допускають отримання рішення, приякому не всі задані об'єкти будуть розміщені в контейнерах. На відміну від них,представлені нижче завдання з класів ODP, CSP і BPP вирішують проблему розміщення всіх об'єктів, при цьому мінімізується або розмір отриманої

упаковки, або число заповнених контейнерів.

Завдання про ранці найбільш часто зустрічаються в наступних областях:

– вантажоперевезення повітряним, морським, залізничним і автомобільним;

– транспортом;

– розкрій матеріалу;

– балансування навантаження в мультипроцесорних розподілених системах;

– рішення задач розподілу ресурсів та завдань теорії розкладів[31];

– розрахунок оптимальних капіталовкладень або формування інвестиційногопортфеля;

– створення криптосистем.

IV. Завдання упаковки в відкриту область («OpenDimensionProblem», ODP)

Клас задач упаковки в відкритій області включає завдання, в якихвикористовуються контейнери, які мають один або кілька змінних габаритнихрозмірів.

Серед завдань упаковки в відкритій області найбільш поширена задачаупаковки на напівнескінченну смугу або завдання рулонного розкрою («StripPackingProblem», SPP) [32], в якій контейнер заданий у вигляді смуги необмеженої довжини і фіксованої ширини. При вирішенні цього завдання мінімізується довжина упаковки,складеної з заданого набору об'єктів. Зокрема, при вирішеннізавдання багатопроцесорного розкладу мінімізується довжина найбільш заповненогоконтейнера серед набору контейнерів необмеженої довжини.

У двомірному випадку зустрічається також завдання мінімізації площіконтейнера, описаного навколо отриманої упаковки.

Для завдання тривимірної упаковки в відкриту область в більшості випадків контейнером є паралелепіпед, що має тільки одну сторону змінної

довжини [33]. Оптимальним рішенням такого завдання буде упаковка всіх об'єктів, що має мінімальну довжину, виміряну вздовж осі, паралельної змінній стороні контейнера.

У завданнях упаковки в відкритій області не завжди використовуються ортогональніоб'єкти. Проблема отримання щільної упаковки нерегулярних фігурних об'єктіввсередині прямокутного контейнера мінімальної площі розглянута в роботах [34].

Завдання упаковки в відкритій області найбільш часто зустрічаються в наступнихобластях:

– розкрій рулонного матеріалу (металу, паперу, тканини та ін.);

– розкрій двомірних листів (з картону, фанери, оргскла, поролону, полікарбонату і ін.) і тривимірних блоків (з дерева, пінопласту, мармуру, граніту та ін.) великої довжини;

– завантаження довгомірних контейнерів при вантажоперевезення;

– проектування компоновок приміщень, обладнання, великихінтегральних схем;

– календарне планування та оптимізація розподілу ресурсів[31];

– підготовка матеріалу для орігамі (пошук найменшого прямокутного аботрикутного контейнера для розміщення в ньому набору кіл аботрикутників).

V. Завдання розкрою («Cutting Stock Problem», CSP)

Завдання цього класу полягають в пошуку найкращого способу розміщення ідентичних об'єктів в контейнерах, які мають фіксовані габаритні розміри. Контейнери можуть бути представлені як одним типом, так і набором різнихтипів, що відрізняються один від одного формою і габаритними розмірами.

Для завдання одновимірногорозкрою (наприклад, розкрою прутків) необхідно знайти спосіб розбиття контейнера заданої довжини на набір об'єктів, слабо відрізняються один від одного по довжині [35].

У двомірному випадку для завдання розкрою в класичній постановці заданий обмежений набір слабо розрізняються між собою прямокутних об'єктів, які повинні бути упаковані в мінімальне число заданих прямокутних контейнерів [36].

Для завдання тривимірного розкрою всі об'єкти також повинні мати схожі геометричні характеристики.

У двомірному і тривимірному випадках розрізняють завдання гільйотини і негільйотинного розкрою [12]. При вирішенні завдання гільйотини розкрою всі об'єкти можуть бути отримані в результаті послідовного виконання наборів наскрізних різців, паралельним кромкам (сторонам) розкраюваного матеріалу. Якщо розкрій не може бути отриманий набором наскрізних різців, то така упаковка називається негільйотинної [37]. На рисунку 1.6 наведені приклади гільйотини і негільйотинного розкрою прямокутного контейнера. Для задач гільйотини розкрою існують алгоритми генерації безвідходних карт [38], які можуть бути використані для аналізу ефективності алгоритмів рішення, що розробляються для вирішення завдань розкрою-упаковки.

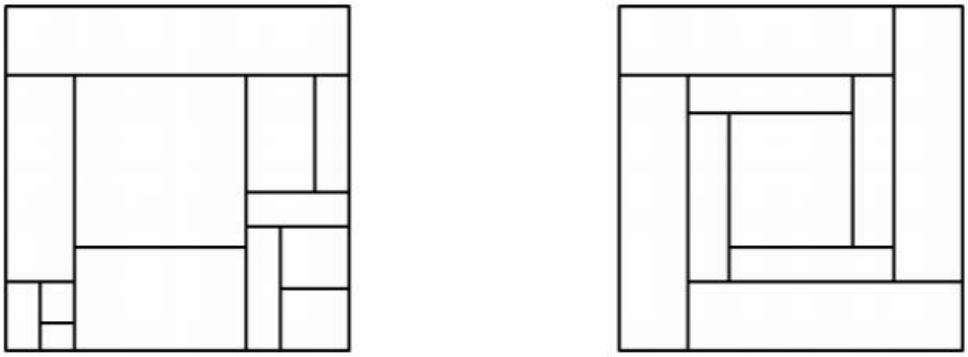


Рисунок 1.6 – Двомірний прямокутний розкрій: а) гільйотинний розкрій; б) негільйотинний розкрій

Завдання розкрою різної розмірності найбільш часто зустрічаються в наступних областях:

– розкрій промислових матеріалів (дерева, металу, скла, шкіри та ін.);

– розподіл оперативної пам'яті і обчислювальних ресурсів в мультипрограмних системах;

– виробництво тонкоплівкових транзисторних рідкокристалічних дисплеїв (розкрій різних підкладок при виготовленні матриць TFT і LCD).

VI. Завдання контейнерної упаковки («BinPackingProblem», BPP)

На відміну від завдань розкрою (CSP), в задачах контейнерної упаковки розміщуються об'єкти повинні мати сильно розрізняються геометричні характеристики. Контейнери можуть бути як ідентичними, так і різнорідними.

Цільовою функцією рішення задачі контейнерної упаковки є функція мінімізації числа використаних контейнерів або функція мінімізація розмірів геометричній області, займаної упаковкою, що складається з усіх об'єктів.

Класична постановка задачі одновимірної контейнерної упаковки («OneDimensionalBinPackingProblem», 1DBPP) має на увазі завдання набору однакових контейнерів заданої довжини і набору об'єктів у вигляді відрізків різної довжини, причому сумарна довжина відрізків не перевищує сумарну довжину контейнерів. Необхідно розмістити всі відрізки в мінімальне число контейнерів. Для завдання одномірної контейнерної упаковки можуть задаватися додаткові обмеження на число об'єктів, що розміщуються в кожному контейнері.

Серед завдань контейнерної упаковки найбільш поширені завдання двомірної прямокутної упаковки (2DBPP – «Two-DimensionalBinPackingProblem») і завдання тривимірної ортогональної упаковки (3DBPP – «Three-DimensionalBinPackingProblem») [39].

1.2.4 Методи рішення задач розкрою. На рисунку 1.3 представлені основні методи, призначені для вирішення різних типів завдань розкрою-упаковки (рисунок 1.7).

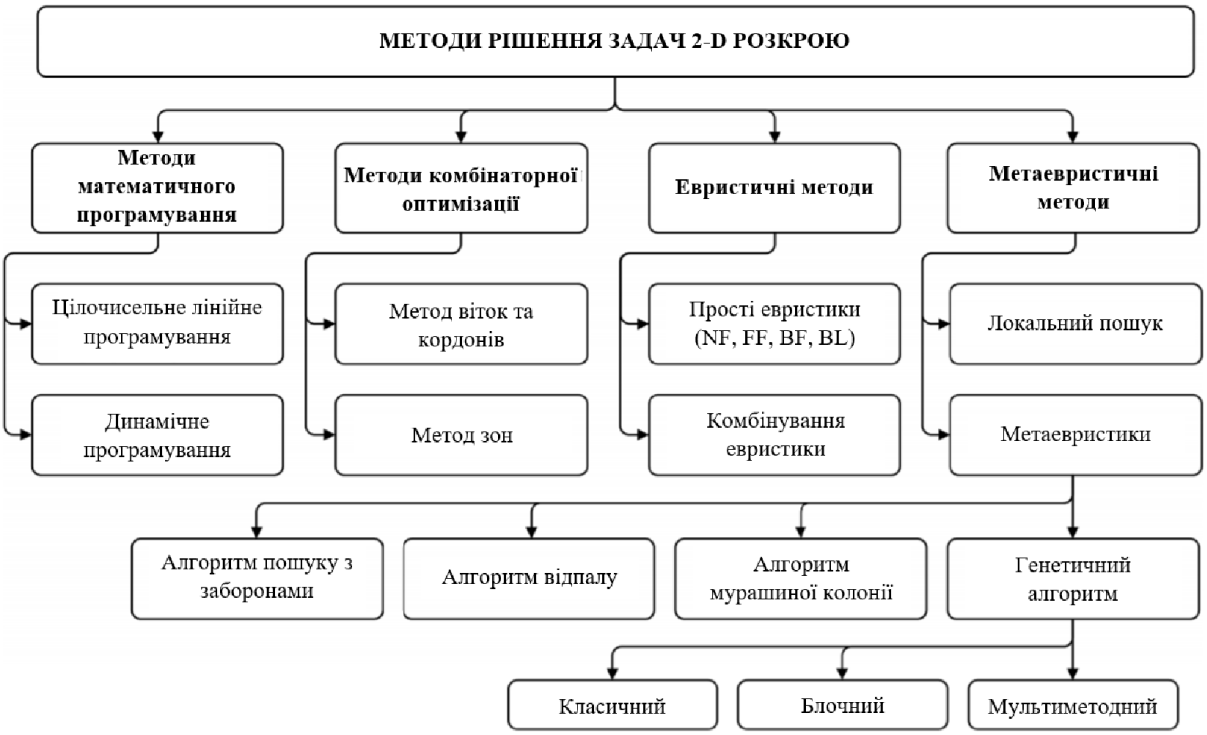


Рисунок 1.7 – Основні методи вирішення завдань розкрою-упаковки

Метод математичного програмування

Серед різних методів математичного програмування для вирішення задачрозкрою-упаковки найбільш популярні методи цілочисельного лінійного ідинамічного програмування.

Методи цілочисельного лінійного програмування для вирішення задачрозкрою вперше застосовані вітчизняними вченими Л.В. Канторовичем і В.А. Залгаллером в 1951 р, при цьому аналогічні методи за кордономз'явилися лише через десять років (P. Gilmore, R. Gomory).

Для задач розкрою-упаковки методи динамічного програмування отрималирозвиток у вигляді ряду спеціалізованих алгоритмів, спрямованих на підвищенняшвидкості виконання завдання розкрою, зокрема, метод переробки списку станів(І. Романовський), алгоритм умовної оптимізації (Е.А. Мухачова), методпошуку мінімуму з заборонами (Е.А. Мухачова). М.В. Ульяновим і О.А. Наумовоїзапропоновано комбіноване алгоритмічне вирішення задачі одномірної упаковки звикористанням методів динамічного програмування, що полягає у виборінайбільш оптимального з декількох алгоритмів в залежності від конкретнихвхідних даних, що проводиться на основі аналізу теоретичних

функційобчислювальної складності алгоритмів [40].

Методи математичного програмування дозволяють найкращим чиномвирішувати завдання одновимірного розкрою і лінійної упаковки, однак при цьому існуютьспособи їх узагальнення для вирішення деяких приватних задач більшої розмірності.Важливо відзначити, що для двомірних задач моделіцілочисельного програмування містять неполіноміальне числозмінних, тому для їх вирішення з використанням методів динамічногопрограмування застосовуються схеми, що перетворюють вихідну задачу в набіродновимірних задач розміщення двомірних об'єктів в смуги, отримані післярозбиття двомірного контейнера [41].

Методи цілочисельного лінійного програмування забезпечують отриманняточного рішення завдань безперервної оптимізації, що робить їх застосовними вумовах серійного і масового виробництва. Багато задач розкрою-упаковкиє завданнями дискретної оптимізації, при цьому їх рішення має місце вумовах одиничного і дрібносерійного виробництв, що обмежує можливостіпрактичного застосування методів математичного програмування. Практичнезастосування методів цього класу виправдано лише для вирішення завдань лінійного ігільйотини розкрою, при цьому для завдань інших типів і розмірностей їх застосуваннянеприйнятне через високу обчислювальну складність.

Метод комбінаторної оптимізації

Завдання розкрою-упаковки об'єктів можуть бути розглянуті в якостіприкладних задач теорії дослідження операцій, що вирішуються комбінаторнимиметодами, заснованих на направленому переборі різних допустимих варіантіврішення задачі [42].

В якості основного методу для комбінаторного рішення оптимізаційнихзадач є метод гілок і меж, який для задач розкрою-упаковки вперше бувпредставлений І.В. Романовським в 1977 р. Його подальший розвиток зобов'язанепояві оптимізованих процедур, спрямованих на скорочення переборуокремих варіантів вирішення.

Загальна схема практичного застосування методу гілок і меж для вирішеннязадач прямокутного розкрою і двомірної ортогональної упаковки запропонована в1986 г. (Ю.Г. Стоян, С.В. Яковлєв).

Загальна проблема різних комбінаторних методів полягає вобмеженості їх практичного застосування для задач великого розміру. Так, длязадач двомірної і тривимірної упаковки, застосування цих методів, як правило,виправдане для наборів, що не перевищують двадцяти об'єктів. Це пояснюєтьсянеобхідністю дослідження всього пошукового простору через неможливістьшвидкого визначення точної нижньої межі завдання. Важливо відзначити, щодоказ оптимальності отриманої нижньої межі також є NP-важкимзавданням, тому окремим напрямком досліджень є визначення нижніхкордонів в завданнях розкрою-упаковки. Наприклад, для визначеннянижньої межі завдання двомірної упаковки в роботі пропонується підхід,заснований на приведенні завдання до одновимірного випадку.

Незважаючи на те, що метод гілок і меж є одним з найбільшефективних комбінаторних методів вирішення завдань розкрою-упаковки, він маєекспонентну трудомісткість, що обмежує його практичне застосування длявирішення завдань розміщення великої кількості об'єктів [43]. Найкращі результатизастосування методу гілок і меж, а також його варіацій для точного рішення задачрулонного розкрою невеликий розмірності (до 29 об'єктів за умови дозволу їхповороту на кут 90° при розміщенні).

Оскільки комбінаторні методи рішення, засновані на повному абочастковому переборі рішень, не реалізовуються при великій розмірності даних, в наш час для задач розкрою-упаковки в умовах одиничного і дрібносерійноговиробництва застосовуються менш трудомісткі імовірнісні, евристичні іметаевристичні методи дискретної оптимізації.

Евристичні методи

Евристичні методи в задачах розкрою-упаковки єоднопрохідні методи, що забезпечують формування компонування на основізаданих правил геометричного конструювання простору упаковки. Огляднаближених

однопрохідних методів, застосовуваних для задач контейнерноїупаковки проведено E.G. Coffman, M.R. Garey i D.S. Johnson [44] в 1984 р

Серед найпростіших евристик розміщення об'єктів найбільш відомі евристикиNF («NextFit» – «наступний відповідний»), FF («FirstFit» – «перший відповідний») iBF – («BestFit» – «кращий відповідний»). Розглянемо ці евристики на прикладізавдання двомірного гільйотини рулонного розкрою при використанні рівневітехнології для формування двомірної упаковки у вигляді набору одновимірних рівнів(смуг). На кожному рівні всі об'єкти розміщуються послідовно один за однимзліва направо, утворюючи рядок, висота якого дорівнює найбільшій висоті серед всіхрозміщених в ній об'єктів. Перед формуванням упаковки об'єктиупорядковуються по спадаючій їх висот.

Евристика NF: вибір для розміщення поточного об'єкта останнього створеногорівня. Якщо цей рівень не може бути використаний для розміщення об'єкта, тостворюється новий рівень (рисунок 1.8, а).

Евристика FF: вибір для розміщення поточного об'єкта першого підходящогорівня. Якщо жоден рівень не може бути використаний для розміщення об'єкта, тостворюється новий рівень (рисунок 1.8, б).

Евристика BF: вибір для розміщення поточного об'єкта найбільш підходящогорівня, для якого невикористане вільний простір після розміщенняоб'єкта буде мінімальним. Якщо жоден рівень не може бути використаний длярозміщення об'єкта, то створюється новий рівень (рисунок 1.8, в).

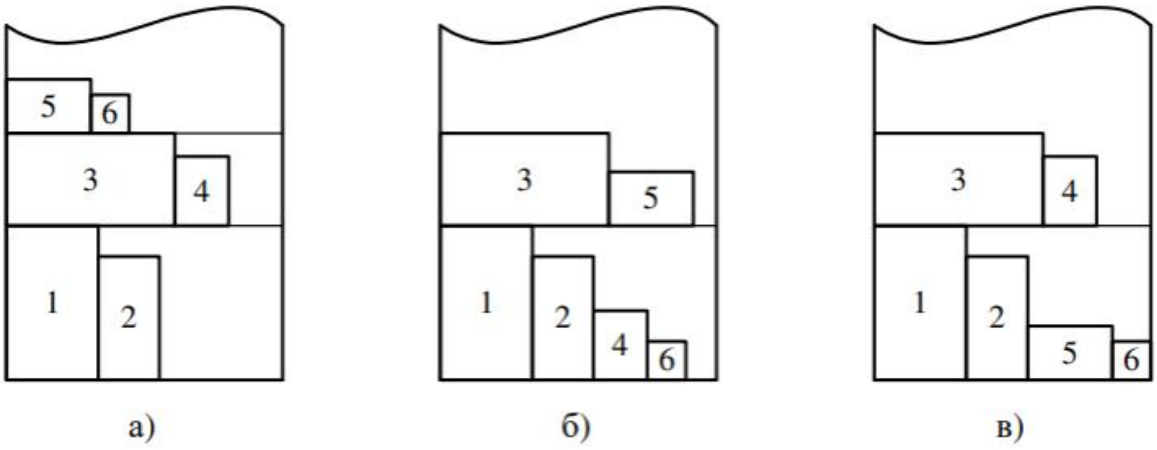


Рисунок 1.8 – Евристики порівневого формування двомірної упаковки: а) евристика NF; б) евристика FF; в) евристика BF

Евристичні методи в задачах розкрою-упаковки часто використовуютьпріоритетні списки, що визначають послідовність вибору об'єктів природміщенні. Так, евристика «нижній лівий» («Bottom Left», BL) використовує упорядкований список об'єктів (по ширині або по висоті) для їхрозміщення в нижній лівий кут контейнера. На рисунку 1.9 наведено дві упаковки,отримані за допомогою евристики BL для набору об'єктів, представлених на рисунку1.9, а. Для упаковки, наведеної на рисунку 1.9, б, що розміщуються об'єкти булипопередньо впорядковані за спаданням ширини (використана послідовністьоб'єктів 3 → 1 → 5 → 2 → 4 → 6).

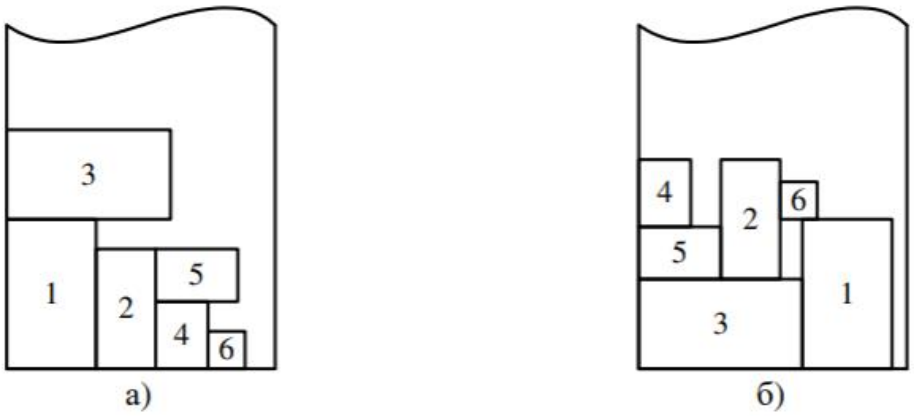


Рисунок 1.9 – Упаковки, отримані однопрохідної евристикою BLпри сортуванні об'єктів:а) по спадаючій висоти; б) по спадаючій ширини

Однопрохідні методи дозволяють отримати швидке рішення задачі великийрозмірності з урахуванням її специфіки, що пояснює їх широку популярність. Ціметоди використовуються, як правило, в якості онлайн-алгоритмів, що застосовуються дляшвидкого формування упаковки [45] і оцінки можливості розміщення об'єктів вконтейнерах.

Подальший розвиток евристичних методів оптимізації пов'язано з їхвикористанням в складі інших алгоритмів оптимізації. Найкращу ефективність однопрохідні евристичні методи демонструють приїх включення в популяцію мультиметодного генетичного алгоритму, в якому генихромосоми представлені однопрохідними евристиками рішення задачі.

Метаевристичі методи

Для підвищення ефективності роботи детермінованих алгоритмів розв'язанняNP-важких завдань застосовуються недетерміновані схеми, засновані навикористанні імовірнісних методів (зокрема, конструктивних методів),а також методів локального пошуку, які здійснюють пошук допустимих рішень воколиці глобального або локального оптимуму [46].

Методи локального пошуку полягають у послідовному змінідеякого поточного рішення в процесі пошуку оптимального або субоптимальногорішення задачі. Оскільки ці методи не здійснюють повний аналіз усієї областізміни цільової функції, то в рідкісних випадках вони дозволяють досягти глобальногооптимуму, однак вони забезпечують швидке отримання локально оптимальнихрішень.

Подальший розвиток методів локального пошуку пов'язано з розробкоюметаевристичних методів оптимізації, які вирішують задачу ітераційногопошуку оптимального рішення на основі застосування недетермінованих процедур,які не потребують повного знання про простір пошуку рішень. На даний моментчасу існує безліч різних класифікацій метаевристичнихалгоритмів оптимізації, при цьому щорічно з'являються нові комбіновані імодифіковані алгоритми, засновані, в першу чергу, на використаннібіоінспированих методів оптимізації.

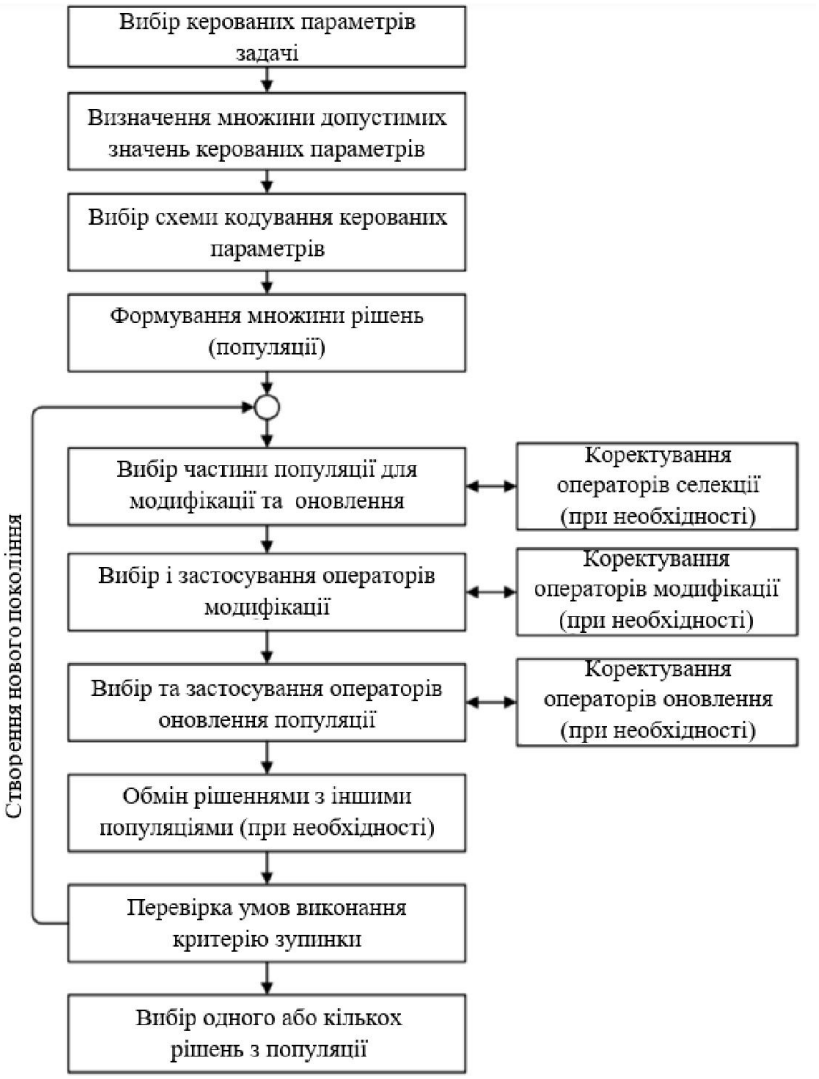


Рисунок 1.10 – Процес рішення оптимізаційної задачі за допомогою популяційного метаевристичного алгоритму

На рисунку 1.10приведена схема, що ілюструє процес застосуваннябільшості метаевристичних алгоритмів, заснованих на оптимізації безлічірішень задачі (т.зв. популяційних алгоритмів). Для популяційногометаевристичного алгоритму необхідно вибрати такі настроюютьсяпараметри:

1. Керовані параметри завдання, найбільш суттєво впливають на якістьоптимізуючого рішення (наприклад, в задачі упаковки об'єктів одним з такихпараметрів є послідовність вибору розміщуються об'єктів);

2. Схему кодування керованих параметрів оптимізації, при цьому

можливовикористання таких кодів:

– двійковий код (найбільш ефективний при вирішенні завдань безперервноїоптимізації);

– натуральні числа (цей спосіб найбільш ефективний при вирішенні задачкомбінаторної оптимізації);

– інші коди (наприклад, код Грея [47]).

3. Схему моделі оптимізації, яка включає наступні параметри:

– правила створення безлічі одночасно модельованих рішень,званого популяцією;

– набір одночасно модельованих популяцій рішень;

– правила вибору підпопуляцій для модифікації рішень, задані у виглядіоператорів селекції;

– вид цільової функції як фактора оцінки якості рішень;

– оператори модифікації (диференціації) і поновлення, що забезпечуютьстворення нових рішень і їх включення в популяції;

– правила зміни наборів застосовуваних операторів модифікації іоновлення;

– критерій зупинки процесу оптимізації.

До числа метаевристичних алгоритмів оптимізації відносять алгоритм пошуку ззаборонами, алгоритм імітації відпалу, популяційні алгоритми, засновані наодночасному використанні безлічі альтернативних рішень (еволюційніалгоритми, алгоритми мурашиної колонії[14], риючи часток, бджолиного рою, штучніімунні системи, алгоритми гармонійного і електромагнітного пошуку та ін.), атакож імовірнісні жадібні алгоритми та інші алгоритми оптимізації, заснованіна багаторазовому застосуванні різних евристичних процедур для дослідженнявсього простору пошуку.

Алгоритм пошуку з заборонами або табу-пошук («TabuSearch», TS), запропонованийF. Glover, заснований на використанні історії пошуку, яка містить перелікдосягнутих локальних оптимумів цільової функції. При цьому в

якостіпоточного рішення використовується рішення, що не потрапляє в околицю жодного зраніше знайдених локальних оптимумів, що містяться в списку заборон. Для цьогоалгоритму реалізовані кілька різних стратегій зміни списку заборон,дозволяють розширити область пошуку оптимального рішення, а також уникнутиповторного перегляду раніше проаналізованих областей.

В основі схеми побудови більшості метаевристичних алгоритмів (в зокрема, генетичного алгоритму та алгоритму відпалу) лежить використаннякінцевих ланцюгів Маркова [48], в результаті чого процес отримання оптимальногорішення є збіжність за ймовірністю, що робить ці алгоритми популярнимидля вирішення різних типів завдань дискретної оптимізації.

Для вирішення NP-важких завдань в даний час застосовується велика кількістьрізних метаевристичних методів оптимізації (засновані, в першу чергу, наеволюційних і популяційних моделях оптимізації), зокрема, алгоритмвідпалу, генетичний алгоритм,алгоритм мурашиної колонії[14], їх гібридні аналоги, і інші алгоритми локальної та глобальної оптимізації.

Найбільшу ефективність для задач розкрою-упаковки демонструютьгенетичні алгоритми, засновані на моделюванні процесу природноїеволюції. Існує безліч різних варіантів реалізації генетичногоалгоритму, при цьому для завдань розкрою-упаковки найбільш часто використовуютьсянаступні варіанти:

– класичний генетичний алгоритм (наприклад, для завдань SPP і 2DBPPрозглядається в роботах D. Liu і H. Teng, H. Gehring і A. Bortfeldt, E. Hopper і B.C.H. Turtoni ін.);

– блоковий генетичний алгоритм, розроблений під керівництвом професора Е.А. Мухачова, заснованийна використанні декодера блокової структури;

– мультиметодний генетичний алгоритм, розроблений професоромІ.П. Норенковим, заснований на поданні рішень задачі у виглядіпослідовності евристик.

Метаевристичні методи забезпечують швидке отримання рішень, близьких до оптимальному, проте не можуть гарантувати потрапляння в глобальний оптимум цільової функції в заздалегідь заданий проміжок часу [49]. Тому проблема підвищення швидкості роботи і якості субоптимальних рішень, одержуваних різними метаевристичними алгоритмами оптимізації, залишається актуальною.

Особливості метаевристичного методу в ГА були розглянуті в тезах LXXIII – International Scientific Conference «EUROPEAN INTEGRATION IN SCIENCE AND INNOVATIONS» [50].

1.2.5 Геометричні аспекти підходів до рішення задач розкрою. В результаті аналізу наукової літератури, присвяченої вирішенню завдань розкрою упаковки, були визначені особливості, пов'язані з геометричним описом контейнерів, що розміщуються об'єктів і формуванням упаковки.

При вирішенні завдань прямокутного розкрою і ортогональної упаковкиконтейнери і розміщуються об'єкти подаються у формі прямокутників іпаралелепіпедів різної розмірності, які можна віднести до класу простихгеометричних фігур. Об'єкти такої форми часто використовуються при вирішенніпрактичних завдань компонування, наприклад, близько 90% приладового обладнаннялітальних апаратів задовільно описується паралелепіпедами.

Для геометричного опису упаковки і подальшого аналізу вільногопростору контейнерів при вирішенні задач прямокутного розкрою і ортогональноїупаковки використовуються:

– набори вузлів, утворених на контурі упаковки і в кутах розміщених об'єктів;

– блок-структури об'єктів, отримані в результаті розбиття упаковки лініями або площинами на рівні, утворені вздовж сторін розміщених об'єктів;

– контур (профіль) упаковки;

– набори областей у формі прямокутників або паралелепіпедів.

Облік геометричних особливостей розміщуються об'єктів являєнайбільшу складність при вирішенні задач фігурного розкрою і компонування об'єктівдовільної форми.

Для завдань фігурного розкрою і упаковки об'єктів складної форми найбільшпоширений спосіб визначення взаємного положення об'єктів з використаннямгодографа функції щільного розміщення, розробленого Ю.Г.Стояном. Цейспосіб використовується при розміщенні об'єктів, заданих в полігональному вигляді.Годограф функції щільного розміщення об'єкта визначає кордон області, якане може бути використана для розміщення цього об'єкта. В процесі пошуку областідля розміщення об'єкта в контейнері виконується моделювання його рухущодо інших розміщених в контейнері об'єктів і визначення точокперетину годографів функцій, складених для розміщується об'єкта і всіхзнаходяться в контейнері об'єктів [51].

Для вирішення завдань розміщення об'єктів з використанням годографа функціїщільного розміщення в харківській школі під керівництвом професора Ю.Г. Стояназапропонований метод Ф-функцій (phi-функцій), який забезпечує надання завданнярозміщення об'єктів у вигляді завдання нелінійного програмування [52]. Привикористанні цього методу для кожного розміщується об'єкта створюються т.зв. Ф-об’єкти, які визначають області допустимого розміщення одного об'єктащодо іншого. Ф-об'єкти створюються з базових простих геометричнихфігур (прямі лінії і дуги кіл для двомірних Ф-об'єктів, а такожплоскі, сферичні, циліндричні і конічні поверхні для тривимірних Ф-об'єктів). Рішення завдання розміщення набору об'єктів полягає у вирішеннісистеми нерівностей з гладкими функціями, які враховують обмеження навзаємне положення об'єктів, розмір якої сильно збільшується зі збільшеннямрозмірності задачі і з уточненням геометричних характеристик розміщуванихоб'єктів. Розвиток цього методу пов'язано з розробкою квазі-Ф-функцій,що забезпечують зменшення розмірності одержуваної системи нерівностей [53].

Важливо відзначити, що для розміщення об'єкта складної форми

потрібнопопередньо декомпозувати його на набір базових простих фігур, для якихіснує відповідна математична модель, що в ряді випадків можепривести до втрати точності опису форми, або розробити для нього новуматематичну модель об'єкта з кусочно-гладкими межами [54], а також всередині контейнерів, що мають обмеження у вигляді набору областей,заборонених для використання при розміщенні об'єктів.

Зважаючи на високу обчислювальну складність, методи нелінійногопрограмування можуть бути використані для розміщення лише невеликого числаоб'єктів. Окремим недоліком методів нелінійногопрограмування, які потребують вирішення системи великого числа нерівностей, єпроблема втрати надійності обчислень при обробці чисел з плаваючоюкоми.

Зазначені недоліки мають місце при використанні полігональногопредставлення об'єктів, однак вони відсутні при використанні вексельноїмоделі, що забезпечує геометричний опис об'єктів і контейнерівдовільної форми і розмірності в вигляді кінцевого набору дискретних елементів(векселів), який в ряді робіт називається також рецепторноюматрицею.

Використання воксельної моделі забезпечує подання об'єктів іконтейнерів у формі ортогональних багатогранників, які з значної частинирозміщуються ортогональних об'єктів. Оскільки швидкість розміщенняортогонального багатогранника в контейнері залежить від числа об'єктів, що утворюютьцей складений об'єкт, то з метою підвищення швидкості формування упаковкинеобхідно виконувати декомпозицію ортогонального багатогранника для отриманнянабору великих ортогональних об'єктів, що займають ту ж область, що і вихіднийнабір ортогональних об'єктів розглянутого ортогонального багатогранника.

Завдання декомпозиції ортогонального багатогранника на безліч великихортогональних об'єктів мінімальній потужності є окремим випадком NP-важкої задачі покриття ортогонального багатогранника

ортогональнимиоб'єктами.

Для декомпозиції ортогонального багатогранника використовуються наступніалгоритми.

5. **Рівневі виділення областей.**Для алгоритму рівневого виділення областей характерний послідовний горизонтальний або вертикальний прохід по всіх точках ортогонального багатогранника. Недоліком реалізації цього алгоритму є залежність швидкості його роботи від габаритних розмірів розглянутого ортогонального багатогранника.

Подібний алгоритм здійснює кілька ітерацій проходу по осередках матриці, яка описує розглянуту область, займану ортогональним многогранником, у вигляді набору рівнів (для горизонтального напрямку об'єднання здійснюється аналіз осередків матриці по рядках, для вертикального напрямку – по стовпцях, для діагонального – як по рядках, так і по стовпцях), з наступним об'єднанням у великі прямокутні об'єкти. Недоліком цього алгоритму є те, що він дає різні результати при використанні різних напрямків об'єднання осередків матриці, при цьому відсутні будь-які відомості про ефективність застосування окремих напрямків об'єднання.

6. **Побудова проекцій в ортогональному багатограннику.**Алгоритм заснований на отриманні ортогональних об'єктів в результаті побудови проекцій граней ортогонального багатогранника на деяку базову площину (підстава). Даний алгоритм має складність O(mlog m), де m – число кутів розглянутого ортогонального багатогранника. Алгоритм реалізований тільки для ортогональних багатогранників в формі так званих гістограм, які представлені у вигляді наборів ортогональних об'єктів, що мають загальну підставу, що істотно обмежує область його практичного застосування.

7. **Відсікання ортогональних об'єктів з наступним об'єднанням**. Дано алгоритм, заснований на відсікання січними площинами ортогональних об'єктів від ортогонального багатогранника з подальшим їх об'єднанням в крупних об'єктів при виконанні проходів уздовж кожної координатної осі. В процесі відсікання використовується кілька січних площин, перпендикулярних

різним координатним осям. Основним недоліком цього алгоритму є поява великої кількості невеликих проміжних ортогональних об'єктів, що уповільнює роботу алгоритму і вимагає виділення додаткових обчислювальних ресурсів на їх обробку. Іншим недоліком цього алгоритму є те, що він є багатопрохідним, і при об'єднанні ортогональних об'єктів вимагає прохід вздовж кожної координатної осі. Цей алгоритм використовується для представлення ортогональних багатогранників, створених з використанням воксельної моделі.

8. **Використання графів**. Алгоритми цієї групи засновані на розбитті складеного об'єкта на ряд прямокутних областей, для яких в подальшому складається граф, визначає їх сусідство один з одним. Рішення завдання мінімізації числа об'єктів, що утворюють ортогональний багатогранник, зводиться до пошуку рішення задачі пошуку графа з найкращими характеристиками.

В результаті аналізу алгоритмів, використовуваних для декомпозиціїортогональних багатогранників, були зроблені наступні висновки:

– більшість алгоритмів орієнтовано на застосування до двовимірним ортогональним многогранників (ортогональним полігонів);

– багато алгоритмів виконують декомпозицію ортогонального багатогранника, для якого обов'язково задається ряд обмежувальних умов (наприклад, наявність загальної підстави для всіх об'єктів, відсутність отворів тощо.);відсутні алгоритми, що забезпечують декомпозицію ортогональних багатогранників довільної розмірності.

1.2.6 Висновки щодо огляду існуючих наукових досліджень.

1. Аналіз наукових публікацій, а також теоретичних і прикладних досліджень показав, що для задач розкрою-упаковки існує сформована в науковому середовищі класифікація, а їх рішення мають велике число практичних додатків в різних областях промисловості і виробництва, що говорить як про актуальності розвитку досліджень в області оптимізації рішень цих завдань, так про і високу затребуваність результатів наукових досліджень.

2. Серед основних методів, що застосовуються для вирішення завдань розкрою-упаковки можна виділити методи математичного програмування, комбінаторної оптимізації, евристичні і метаевристичні методи. методи математичного програмування забезпечують можливість отримання рішення задачі із заданим точністю (в тому числі оптимального для завдання лінійного програмування), проте їх застосування обмежене лише невеликими розмірами завдання (як правило, до 20 об'єктів) з огляду на їхню високу обчислювальну складність. методи комбінаторної оптимізації, засновані на виконанні часткового перебору різних варіантів рішення задачі, також є дуже чутливими до розмірності задачі. Найкращий компроміс між швидкістю рішення і якістю рішення в умовах одиничного і дрібносерійного виробництва досягається при використанні евристичних і метаевристичних методів оптимізації. серед різних метаевристичних методів оптимізації для задач розкрою і упаковки найвищу ефективність демонструють алгоритми, засновані на використанні еволюційних моделей оптимізації. Евристичні методи можуть бути використані для виконання однопрохідної оцінки можливості розміщення об'єктів, а також для їх застосування в складі інших оптимізаційних моделей.

3. Рішення задачі найкращого розміщення об'єктів на основі евристичних і метаевристичних алгоритмів оптимізації може бути зведено до пошуку послідовності об'єктів, розміщення яких відповідно до деякого використовуваним правилом або евристикою забезпечує отримання найбільш щільною упаковки, що дозволяє використовувати загальні оптимізаційні схеми при розміщенні об'єктів, що мають різні геометричні характеристики.

4. Для підвищення швидкості формування упаковки з використанням моделі необхідно виконувати декомпозицію розміщуються ортогональних багатогранників на безліч великих ортогональних об'єктів. Існуючі алгоритми не дозволяють вирішувати задачу декомпозиції ортогонального багатогранника довільної розмірності.

1.3 Аналізіснуючихдодатків для задач розкрою в умовах фіксованих 2-Dобмежень

Для визначення ключових характеристик веб-сервісу, що розробляється, був проведений аналіз існуючих систем і визначені їх особливості та недоліки.Розглянемо кілька прикладів вже готових додатків розроблених для роботи з розкроєм матеріалів в умовах фіксованих 2-D обмежень та проаналізуємо їх.

Приклад 1. Програма «Об'ємник – меблеве підприємство», виконана в світлих тонах, містить графічний редактор та дає можливість розкроїти листовий матеріал.

У програмі легко спроектувати які завгодно меблі. Досить перенести складові з правої панелі до робочого простору. Крім того, ви можете вручну переміщати шафи, використовуючи функцію зміщення. Згодом через 15-20 хв. роботи ви зрозумієте, що все досить зручно та інтуїтивно зрозуміло.

У програмі є велика основа класичних меблів: кухні з масиву дерева, пластику, привабливі скручені фасади, каркасні профілі, всілякі шафи, підвісні, ортопедичні фундаменти. Можливо зробити свій ескіз або застосувати моделі які пропонує виробник.

У результаті документ зберігає в БД заявок, але в програмі є CAD-меню, яке розширює можливості заощадження та установки 3D-моделей: STL, IGES, VRML, STEP.

Після створення меблевого плану деталі представлені для різання. Доступні різні режими розрахунку для роботи з лишками ДСП, регулювання товщини ріжучої деталі, регулювання поперечного або поздовжнього різу [55].

Недолік – відсутність можливості зберігати «залишки» роботи проекту та використати їх повторно.

На рисунках 1.11 – 1.13 зображений інтерфейс додатку«Об'ємник – меблеве підприємство».

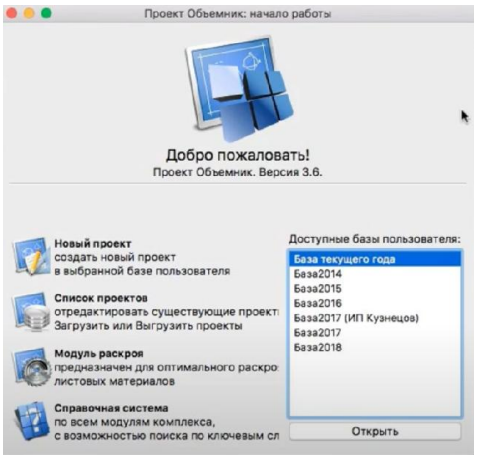


Рисунок 1.11 – Початок роботи з програмою «Об'ємник – меблеве підприємство»

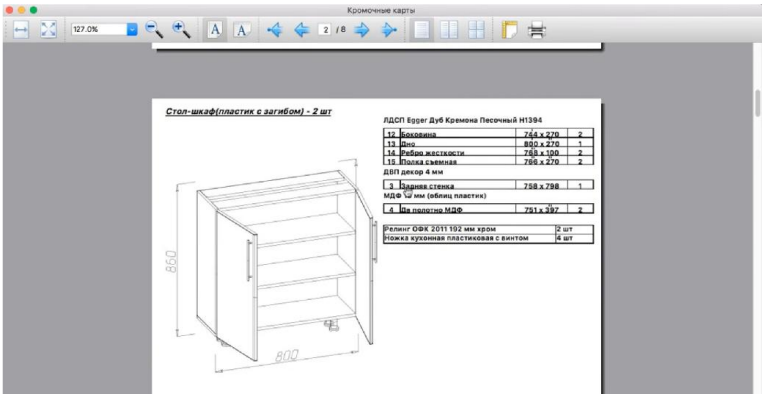


Рисунок 1.12 – Кромкова карта розробленого проекту в програмі«Об'ємник – меблеве підприємство»

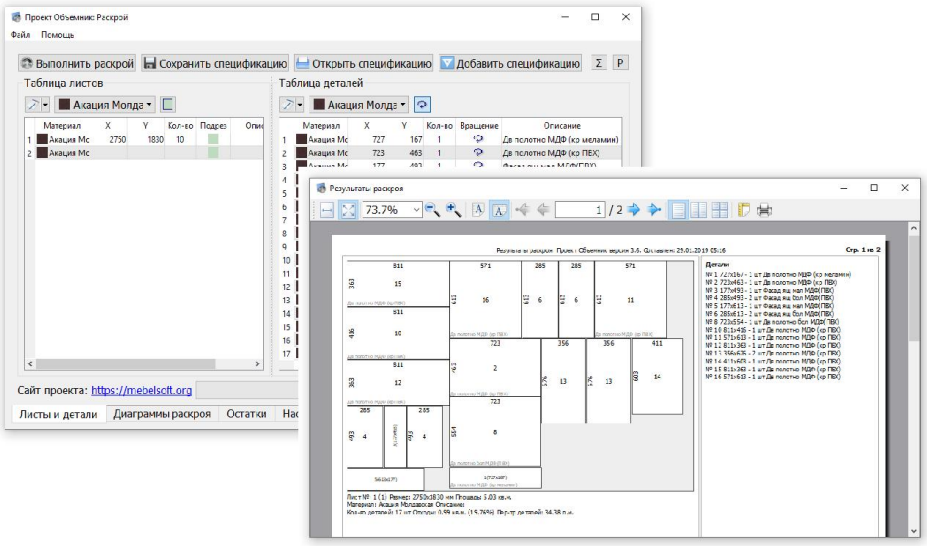


Рисунок 1.13 – Таблиця розміру деталей та схема розкрою в програмі«Об'ємник

– меблеве підприємство»

Приклад 2.Програма «Астра Розкрій», виконана в білих тонах, має можливість імпорту карт розкроїв з інших форматів, швидкий ввід інформації, високоточний розкрій, зручний інструмент для внесення змін в розкрійні карти вручну, аудит на підприємстві та можливість використати обрізки з виробництва.

«Астра Розкрій» можливо застосувати як в офісі, так і на сервері. Встановлюючи додаток на сервері локальної мережі вашої організації, додаток можна запустити на кожному комп'ютері в мережі. Для завантаження на сервер в інтернет надається безкоштовна програма різу, яка може бути розташована в системі прийому заявок онлайн.

Є можливість вводити розміри деталей вручну, обирати їх з наявних на складі товарів, вводити дані з таблиці Excel або текстового файлу. Центри обслуговування, що обслуговують виробників меблів, знадобиться, між іншим, функція поєднання замовлень, включаючи їх угруповання, що дає деяку економію матеріалів та часу.

Автоматизоване різання матеріалів проходить по технічним вимогам виготовлення. Показники дають змогу задати: обріз країв листа, ширину зрізу, образ зрізу, максимальну ширину зрізу лінії тощо.

Найкраща ефективність також досягається з допомогою можливості ручного розташування листів: центрування, зсунення, злипання компонентів тощо. Є можливість скасування вироблених деталей та збільшення крою.

Прорахунок відходів після процесу вирізання деталей можливо виконати як автоматизовано так і вручну. Працюючи з списком обрізків, є можливість редагування списку за якостями чи видаляти їх.

Даний додаток являється еталоном для мене.

На рисунках 1.14 – 1.19 зображений інтерфейс програми «Астра Розкрій».

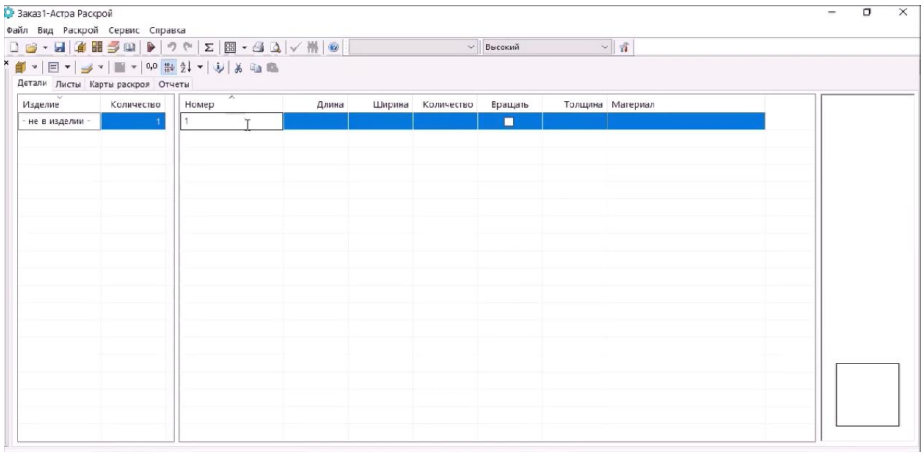


Рисунок 1.14 – Початок роботи з програмою «Астра Розкрій»

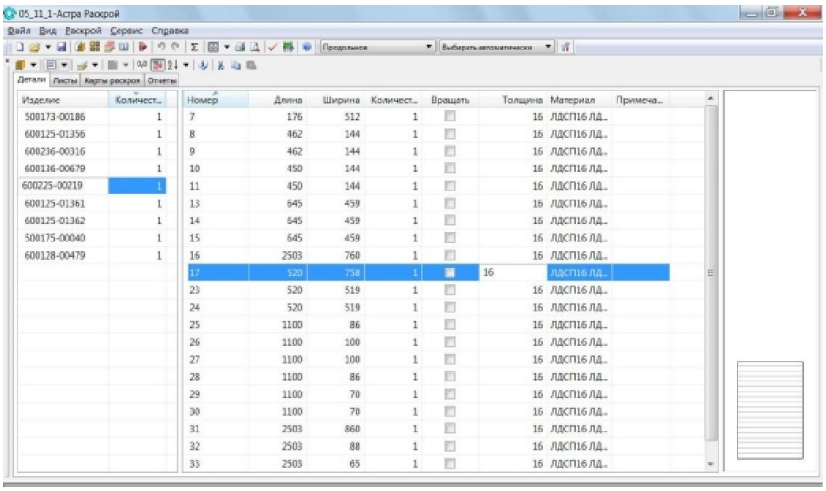


Рисунок 1.15 – Формування замовлень в програмі «Астра Розкрій»

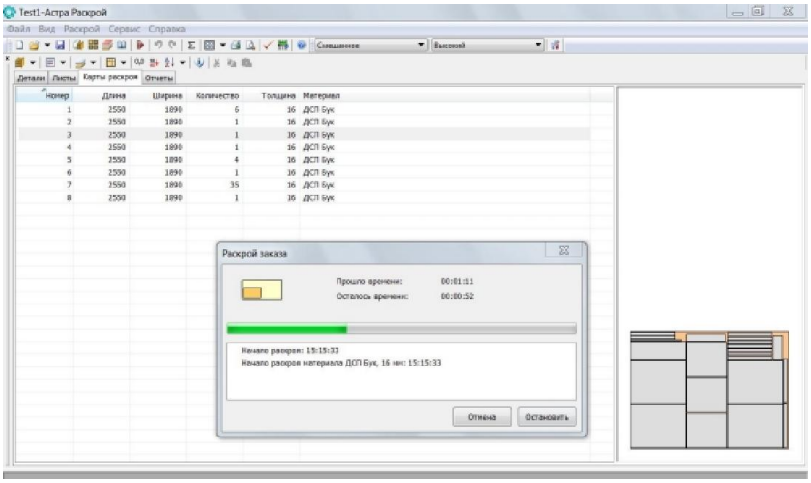


Рисунок 1.16 – Прорахунок оптимального розкрою в програмі «Астра Розкрій»

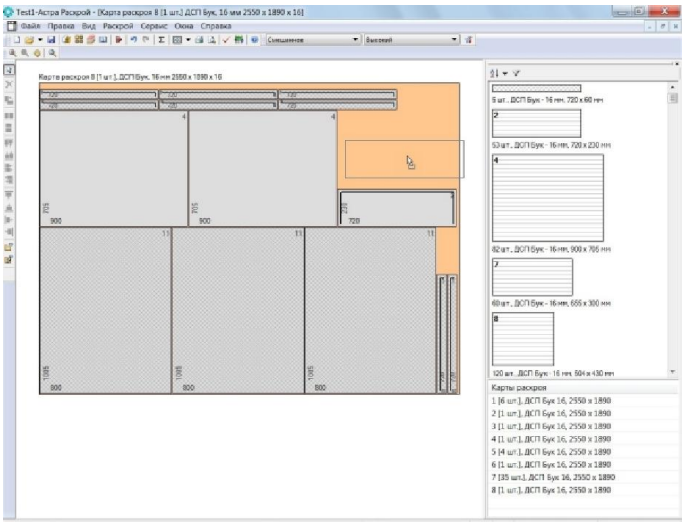


Рисунок 1.17 – Ручне редагування карт розкрою в програмі «Астра Розкрій»

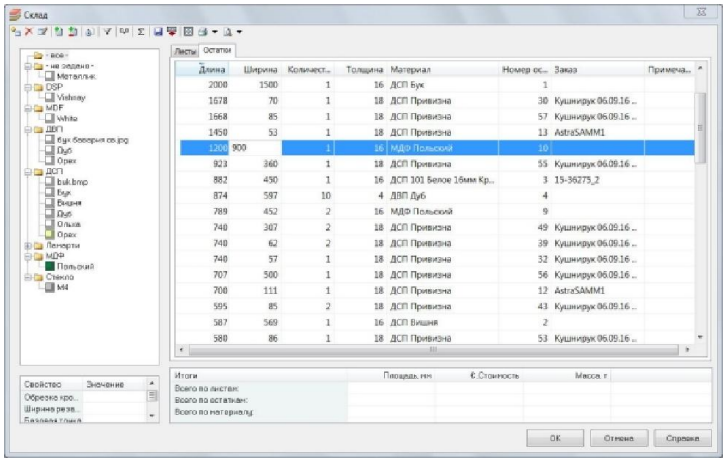


Рисунок 1.18 – Повний облік залишків після розкрою в програмі «Астра Розкрій»

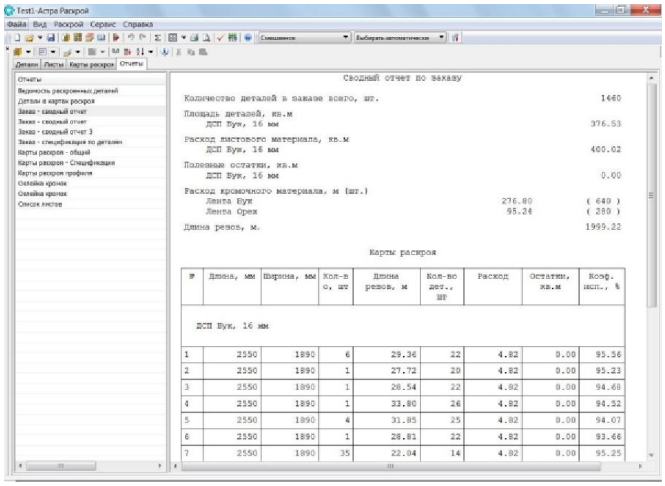


Рисунок 1.19– Обширний автоматично сформований звіт в програмі «Астра Розкрій»

Приклад 3.Програма «Catting 3» виконана в світлих фіолетових тонах та має декілька варіацій тем на вибір. Вона використовується для розкрою матеріалу на прямокутні деталі. В основі лежить алгоритм, що дає змогу за короткий час зробити розкрій деталей на матеріалі з мінімальними лишками.

Існує можливість визначити випадкову кількість сторінок та фрагментів для вирізування, одночасна вказівка декількох текстур, що підлягає різанню. Програма може працювати в кількох режимах, збереження бази матеріалів та лишків, транспортування вирізаних частин з однієї ріжучої сторінки на іншу, позначення лишків, позначення країв, перенесення деталей з AutoCAD та схожих програм, передача властивостей та результатів різання до схожих програм, транспортування різальних листів, запис та відновлення результатів різання, застосування тем, прорахунок загальної площі деталей, довжина різання, ділянки залишків, відходи тощо, захист зазначених сторінок та деталей, таких як їх специфікація

Винятковий, швидкісний алгоритм, що лежить в основі програми, дає можливість легко розрізати з мінімально можливими відходами. Користувач отримує багато панелей та деталей для розробки. [57].

Недоліком на мою думку є інтерфейс, він не є інтуїтивно зрозумілим для першого використання, та виявився досить складним на етапі додання нового листа матеріалу. Програма не має можливості автоматично створити звіти щодо замовлення.

На рисунках 1.20 – 1.22 зображений інтерфейс програми «Catting 3».

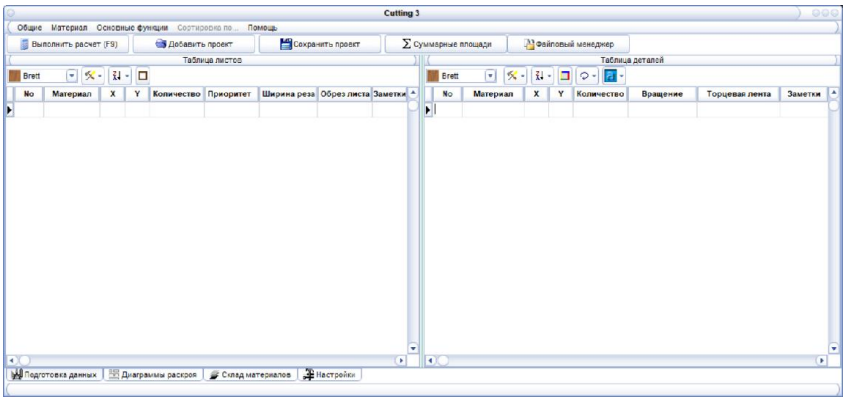


Рисунок 1.20 – Початок роботи з програмою «Catting 3»

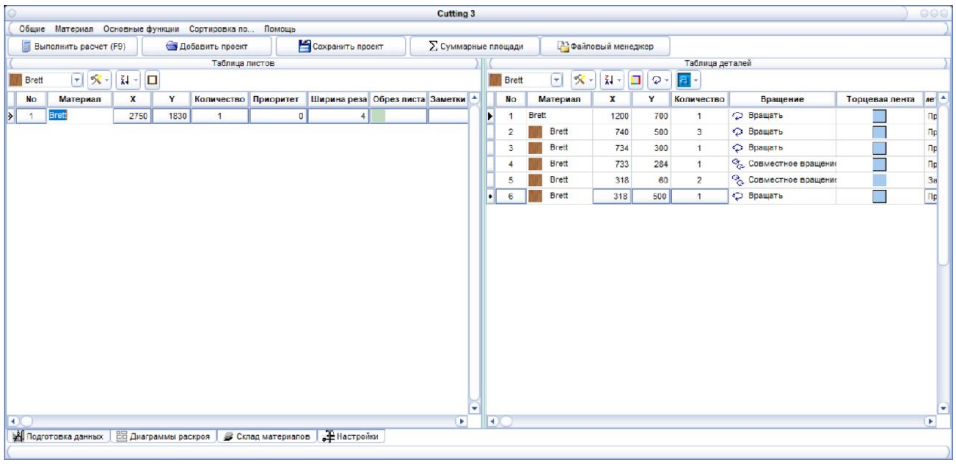


Рисунок 1.21 – Введення даних для роботи з проектом в програмі «Catting 3»

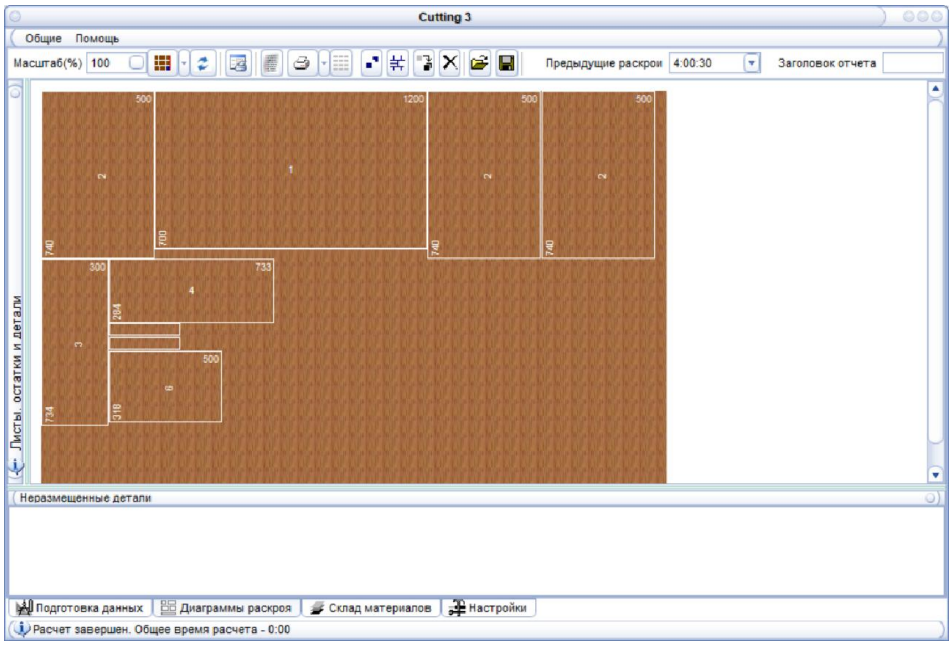


Рисунок 1.22 – Схема розкрою деталей в програмі «Catting 3»

Приклад 4.Програма «Sawyer» виконана в переважно в сірих тонах. Дає змогу розкраювати меблеві щити для виробництва, зробити автоматичним поріз меблевих деталей, проводити аудит складу щитів і комплектуючих. Додаток призначений для меблевих виробництв і невеликих компаніях. Sawyer вбудована в пакет Woody [58]. Все що створено з використанням даного пакету, можливо використати в програмі Sawyer для майбутньої розробки.

В інтернеті можливо завантажити візуальні редактори, що мають функцію розкрою, розпилювання, контролю над розмірами заготовок. До них

слід зарахувати програму Sawyer. Її основні можливості:

– на основі спроектованих виробів та інформації про матеріальні складські залишки складання карти розкрою;

– ручне форматування карт розкрою з роздруківкою їх креслень;

– створення карт розкрою з урахуванням ширини та послідовності розпилу, технологічних відступів та текстури;

– ведення статистики сумарної площі залишків та відсотка відходів.

Загалом програма забезпечує оптимізацію розкрою та підтримку обліку складських матеріалів незалежно від профілю, а також розмірів підприємства. Головна її перевага – забезпечення ефективного алгоритму оптимізації використання матеріальних ресурсів [59].

Недоліком на мою думку є інтерфейс, він досить застарілий. Відсутність можливості зберігати «залишки» роботи проекту та використати їх повторно. Відсутність оновлень.

На рисунках 1.23 – 1.25 зображений інтерфейс програми «Sawyer».

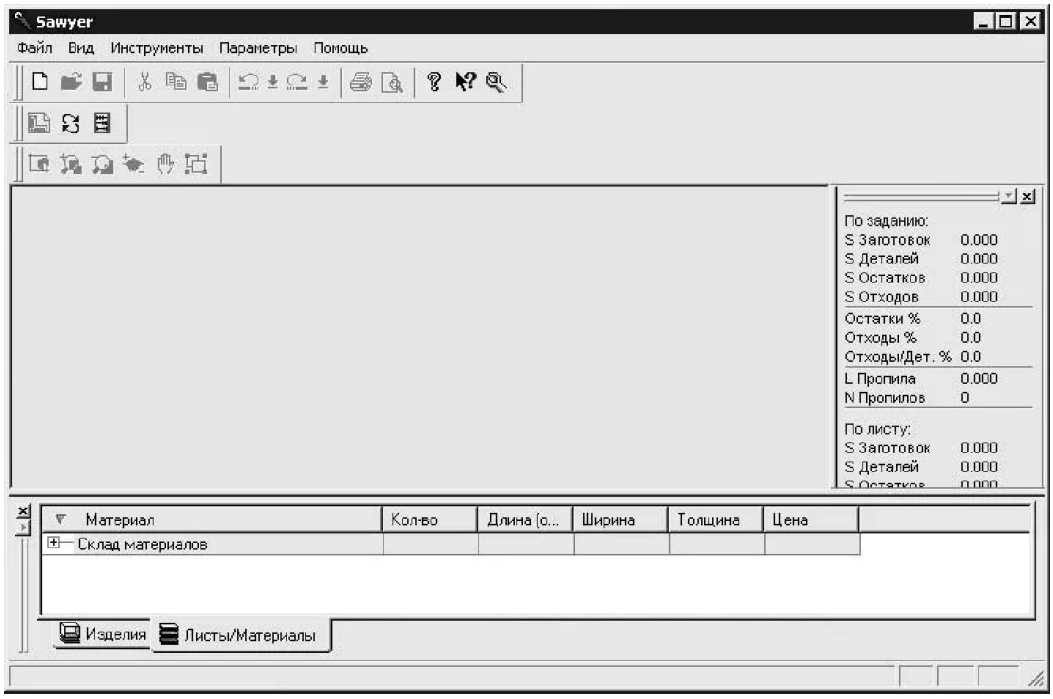


Рисунок 1.23 – Початок роботи з програмою «Sawyer»

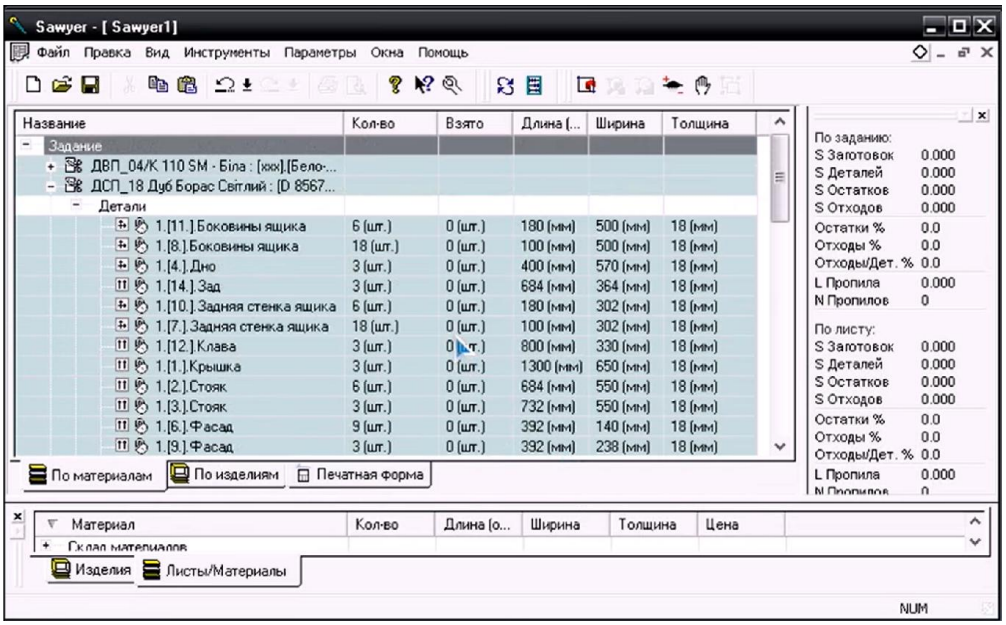


Рисунок 1.24 – Підготовка даних в програмі «Sawyer»

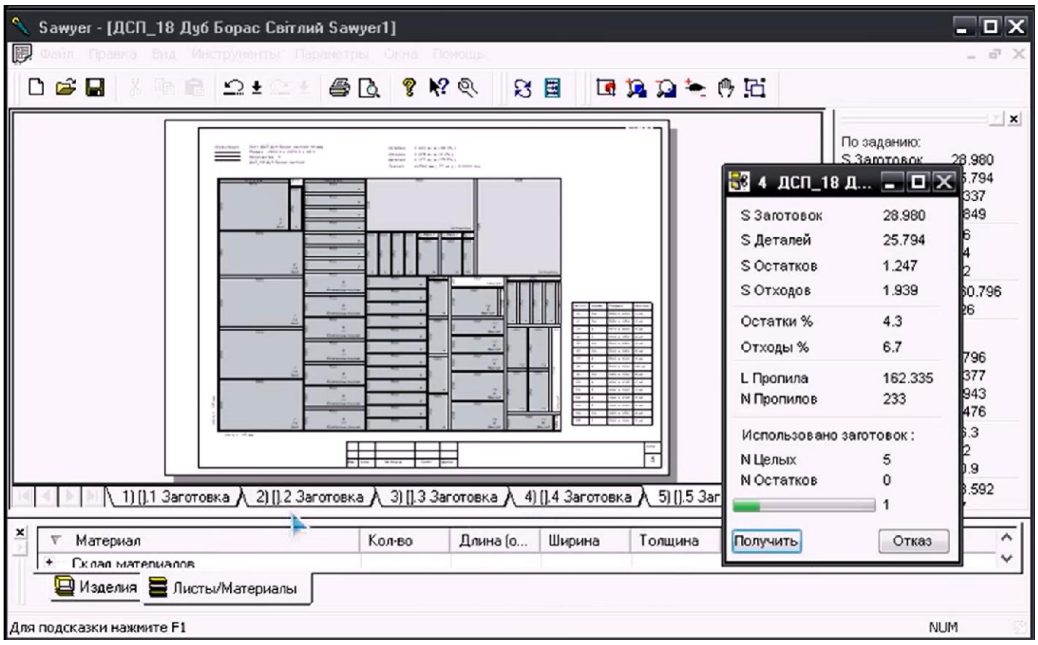


Рисунок 1.25 – Схема розкрою деталей в програмі «Sawyer»

На підставі проведеного аналізу були сформульовані основі вимоги до додатку:

– можливість додавання деталей з фіксованим розміром;

– можливість додавання нових листів матеріалу їх кількість та ціну за погонний метр;

– вивід діаграми оптимізованого розкрою матеріалу на основі

генетичного алгоритму;

– прорахування сумарної площі листа матеріалу, деталей та відходу;

– прорахування ціни використаного матеріалу;

– автоматичне формування звіту про результати роботи над проектом;

– можливість перегляду короткої довідки щодо роботи з додатком.

1.4 Функціональні вимоги

Завданням даної роботи є web-додаток на базі генетичного алгоритму для задач розкрою в умовах фіксованих 2-D обмежень, для компаній які займаються виробництвом корпусних меблів. Для того, щоб визначити функціонал модулів розглянемо функції, які можуть виконувати цільові особи додатку.

Таблиця 1.1 – Функції користувача в модулі

Функція	**Виконавець в АСУ**
РЕДАГУВАННЯ. Створення та редагуваннянових листів матеріалу.	Користувач
РЕДАГУВАННЯ. Створення та редагування деталей для розкрою.	Користувач
ПЕРЕГЛЯД. Перегляд діаграми оптимізованого розкрою матеріалу на основі генетичного алгоритму.	Користувач
ПЕРЕГЛЯД. Перегляд прорахунку сумарної площі листа матеріалу, детале, відходу, а також ціну за використаний матеріал.	Користувач
ПЕРЕГЛЯД. Перегляд звіту про результати роботи з проектом.	Користувач
ЕКСПОРТ. Експорт звіту про результати роботи з проектом.	Користувач

РОЗДІЛ2
ПРОЕКТУВАННЯ WEB-ДОДАТКУ НА БАЗІ ГЕНЕТИЧНОГО АЛГОРИТМУ В УМОВАХ ФІКСОВАНИХ 2-DOБМЕЖЕНЬ

2.1 UXcaseдослідження особливостей додатку

Проектом є діяльність команди, спрямована на певний результат. Коли результат (мета проекту) досягається, проект вважається завершеним. Існують деякі види діяльності, які є проектами, наприклад, підтримка користувачів чи супровід товару. Такі види діяльності ми називаємо "Активністю".Кожен проект чи активність характеризуються:

– метою, заради якої цей проект ініційований, або набором вимог, що входять від користувачів;

– командою, тобто у кожного проекту є склад учасників, які мають доступ до проекту та його артефактів;

– процесом, яким здійснюється управління проектом;

– результатами (або проектними артефактами), що утворюються внаслідок роботи проектної команди.

Проекти дають змогу ізолювати команди (або проектні артефакти). Однак іноді потрібно бачити дані всіх проектів або якогось підмножини проектів. Включати керівників у кожен із проектів - це незручно. Для цього можна використовувати програми та портфелі, які дозволяють об'єднувати дані різних проектів на одному екрані [60].

Ітеративна розробка ПЗ - це процес створення програмного забезпечення, який здійснюється невеликими етапами, в ході яких ведеться аналіз отриманих проміжних результатів, висуваються нові вимоги та коригуються попередні етапи роботи.

Життєвий цикл проекту при ітераційній розробці розбитий на послідовність ітерацій, кожна з яких, по суті, є проектом в мініатюрі, тобто включає всі процеси розробки ПЗ (збір і аналіз вимог, складання специфікацій,

безпосередню реалізацію, тестування і запуск), але у межах однієї ітерації розробляється не весь проект, лише його версія чи окрема частина.

Як правило, мета кожної ітерації –це отримання версії ПЗ, що включає як нові або перероблені можливості, реалізовані в ході поточної ітерації, так і функціональність всіх попередніх ітерацій. Результат фінальної ітерації містить усю необхідну функціональність продукту.

Бюджет і терміни, необхідні реалізації фінальної версії зазвичай спочатку не встановлюються, оскільки визначається загальний обсяг робіт і вимоги формуються у процесі реалізації.

Ітеративність (iteration, «повторення») в даному випадку означає підхід, заснований на виконанні завдань в рамках «міні-проектів», інкрементність (increment «збільшення») означає послідовне додавання функціоналу до продукту, що розробляється, а еволюційність (evolutio, «розгортання») –процес розвитку продукту, що нагадує природний розвиток біологічних видів[61].

Саме тому для проектування даного проекту було обрано ітеративну модель розробки. На рисунку 2.1 – зображена візуалізація ітеративної моделі розробки.

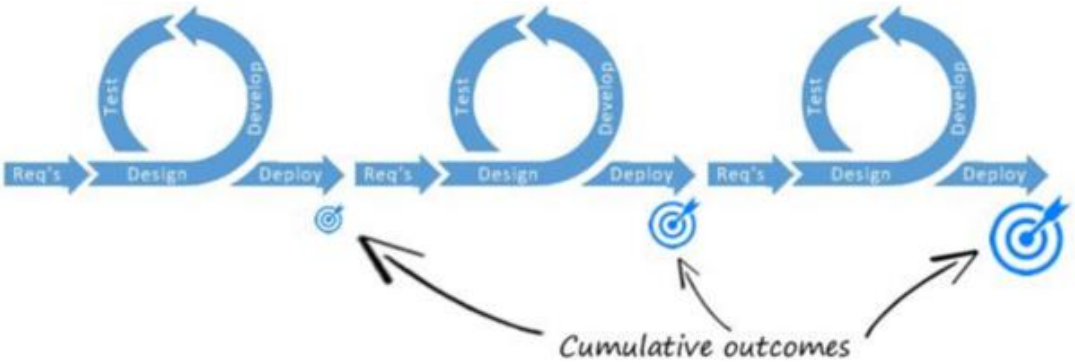


Рисунок 2.1 – Ітеративна модель розробки.

Перш ніж розпочинати розробку, необхідно визначити весь процес, скласти план заходів і скласти розрахункові терміни виконання. Хороший план дій є одним із компонентів успіху проекту, а також допомагає побачити загальну картину проекту з кінцевими цілями та ключовими етапами його

виконання.

Основними розділами є:

– дослідження;

– ідея;

– розробка;

– впровадження.

На рисунку 2.1 – зображено план активності проекту.

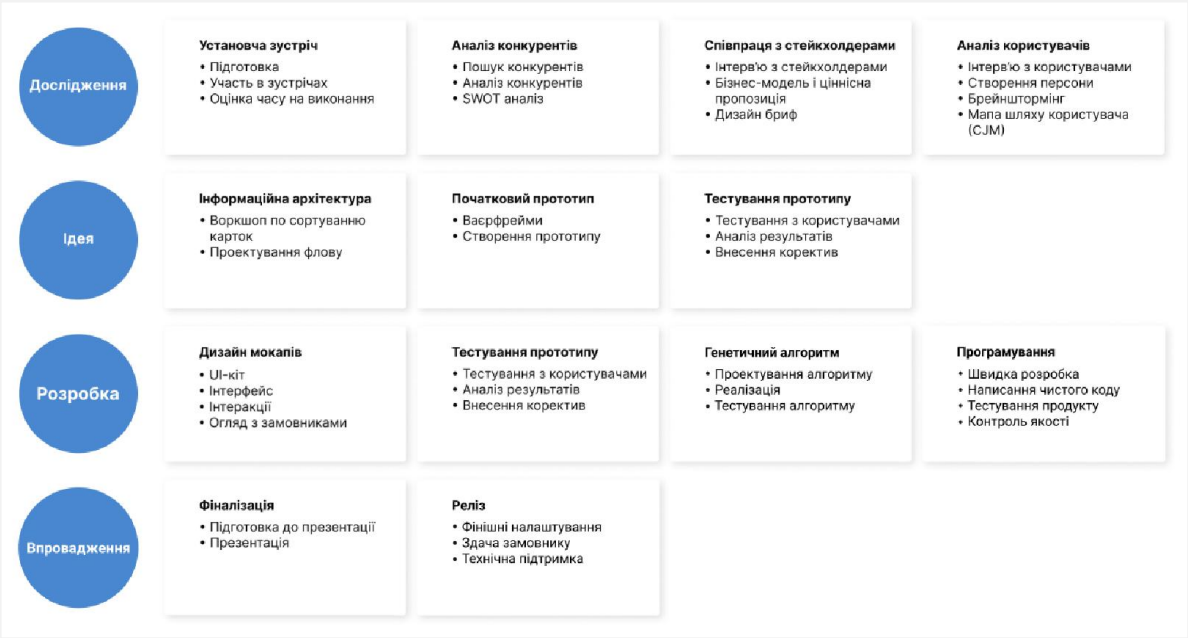


Рисунок 2.1 – План активності проекту.

Важливою частиною всього процесу було знайомство зі стейкхолдерами, щоб дізнатися про подробиці проекту та поняття його інтересів, цілей та потреб.

Було важливо розуміти, які цілі має проект, хто та які завдання має виконувати продукт. Для цього було проведено інтерв'ю з клієнтами.

Підготовка – важлива частина успіху, тому раніше було написано опитування з питаннями. Ось деякі з них:

1. Якою є основна думка проекту? Яку проблему вирішує?
2. Хто користуватиметься послугою?

3. Які переваги для користувачів?

4. Які технології передбачені у проекті?

5. Як виглядає успіх продукту та які показники ефективності визначають цей успіх?

6. Як це буде працювати, джерела фінансування підтримки та розвитку продукції тощо.

Після проведення співбесіди та аналізу відповідей було створено резюме з дизайну.

Наступним кроком було проаналізувати прямих та непрямих конкурентів майбутнього продукту. Проведення такої діяльності дозволило краще зрозуміти ринок.

На рисунку 2.3 – зображено дослідження ринку з додатками-аналогами.

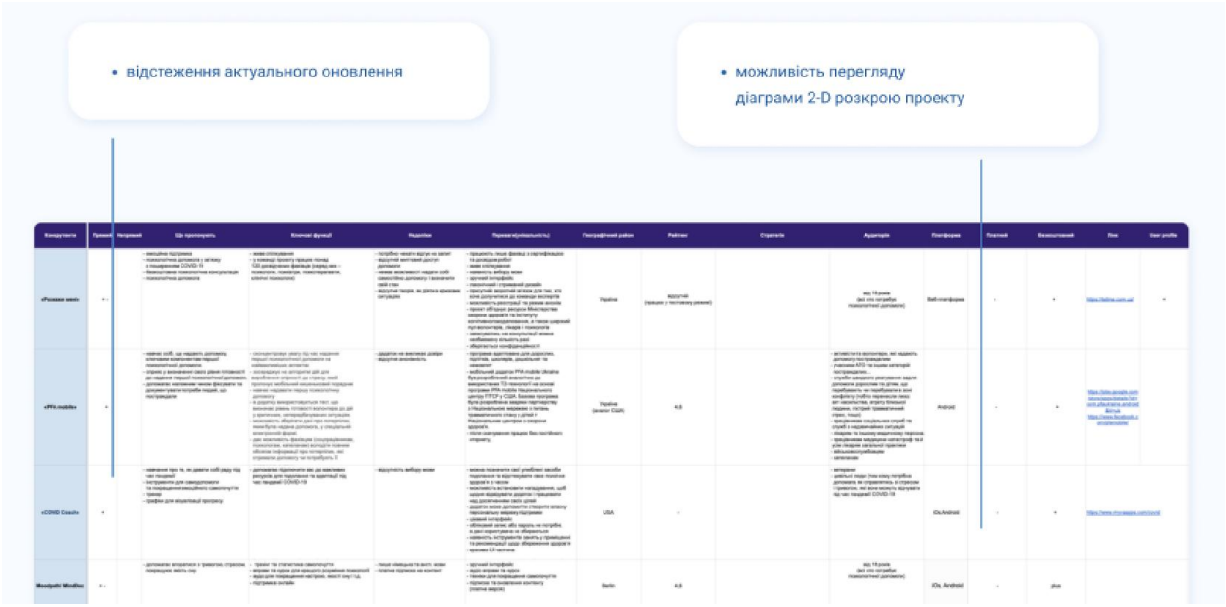


Рисунок 2.3 – Результати дослідження конкурентних аналогів.

Після проведених кількісних та якісних досліджень ми почали створювати особистість. Людина – це узагальнене визначення цільової аудиторії. Створення особи допомогло нам у процесі розробки дизайнерських рішень для майбутніх користувачів.

На рисунку 2.4 – зображено персона користувача.

Рисунок 2.4 – Портрет користувача додатку.

Наступним важливим кроком було створення CJM (CustomerJourneyMap).

CustomerJourneyMap (у перекладі – карта шляху клієнта) – це шлях до покупки з погляду клієнта. Зазвичай це таблиця чи інфографіка, де наочно показані етапи дозрівання клієнта. На кожному етапі докладно описані точки контакту з компанією, дії, думки, емоції, проблеми клієнта, а також чого не вистачає для ухвалення рішення. У сфері розробки ПЗ такий шлях називається userjourney (подорож користувача, наприклад, на сайті).Чим краще ви знаєте свою аудиторію та розумієте її запити, тим легше вам підлаштувати свій бізнес до її побажань та зробити якісний продукт[62].

Тому було сформовано CJM Щоб візуалізувати весь шлях користувача та пояснити його взаємодію з майбутнім продуктом у всіх точках. На рисунку 2.5 – зображенастворена CJM.

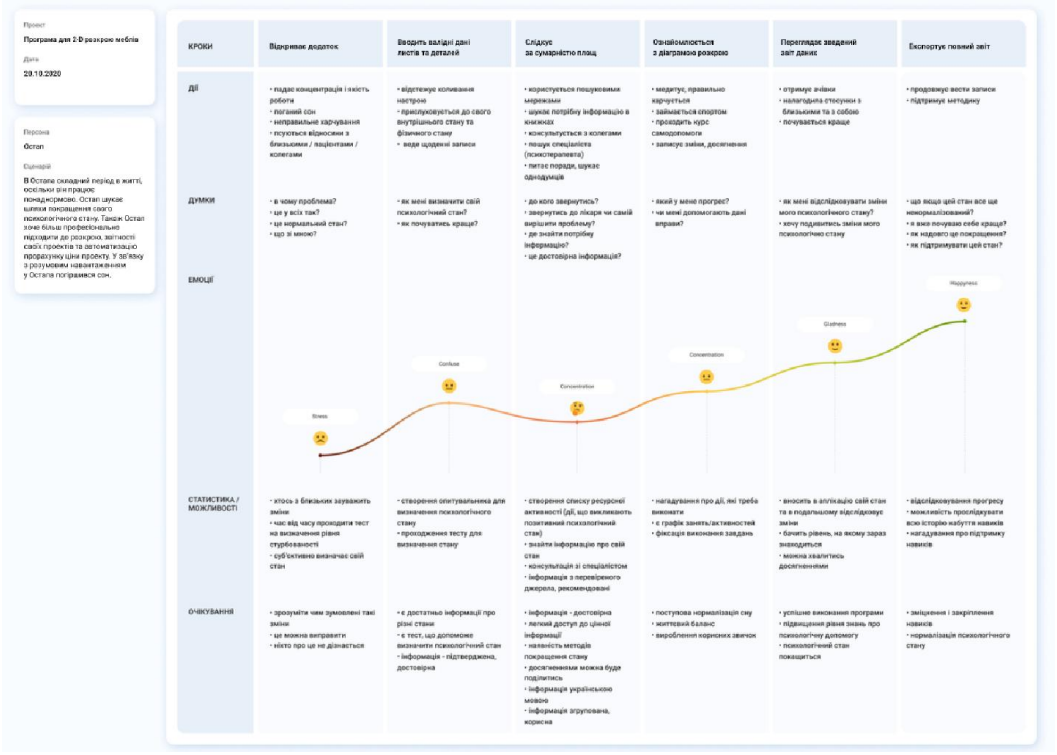


Рисунок 2.5 – CustomerJourneyMap.

Наступним етапом було формування плану активності проекту (MindMap). Для реалізації плану було обрано програму Mindomo[63].

Метод MindMap – техніка візуалізації, яка допомагає вивчити нову інформацію або вирішити завдання, що стояло перед нами. Якщо звичні методи запам'ятовування нової інформації та планування здаються Вам застарілими, пропонуємо познайомитись із ментальними картами (MindMap).

Якщо з традиційними способами структурування та відображення інформації все зрозуміло, то карта думок – це щось нове. MindMapping – це спосіб представлення інформації за допомогою діаграми зв'язків. Суть інтелект-карти – побудувати асоціативні ланцюжки. В основі карти розуму потрібно поставити основну думку, а далі малювати гілки, розбираючи загальну ідею більш докладні. Скетчноутинг – це теж своєрідна побудова ментальних карт. Для даного проекту було виділено наступні ключові позиції:

– загальні відомості;
– генетичний алгоритм;
– підготовка даних;

– сумарні площі;

– діаграма розкрою;

– зведений звіт;

– довідка.

На рисунку 2.6 – зображено план активності проекту.

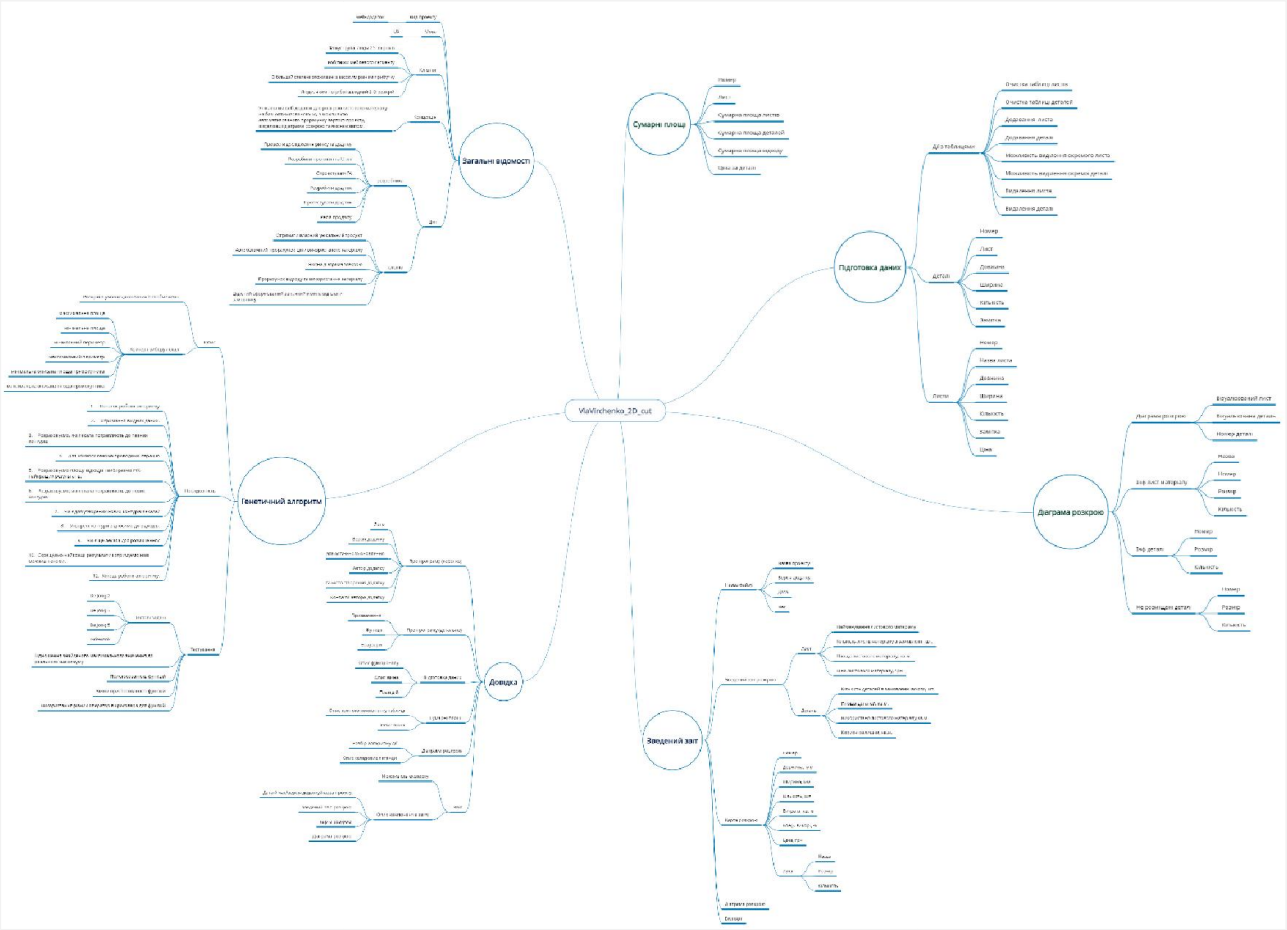


Рисунок 2.6 – План активності проекту.

Більш детальний Outlineпроекту можна переглянути в Додатку В.

LeanCanvas– це шаблон для побудови бізнес-моделі. В основі моделі – філософія бережливого мислення та методологія Lean Startup1. Вонадопомагає продуктовому менеджеру або власнику продукту швидко описати свій задум.

Зміст шаблону можна змінювати в процесі справи. Це зручно для гнучких процесів коли спочатку незрозуміло, що вийде. З його допомогою легко зібрати інформацію про майбутній продукт на одній сторінці, де всі наявні гіпотези

будуть під рукою. Особливостями Lean Canvasє:

– допомагає творцю та менеджеру зрозуміти продукт з усіх боків;

– допомагає творцю та менеджеру пояснити свої ідеї іншим.

Після того, як усі основні дані були зібрані, було вирішено провести семінар із клієнтом, щоб краще зрозуміти цілі та плани підприємства. Через пандемію семінар проводився онлайн. Було обраноMiro[64] як основне середовище для проведення воркшопів.На рисунку 2.7 – зображенаствореналean Canvas.

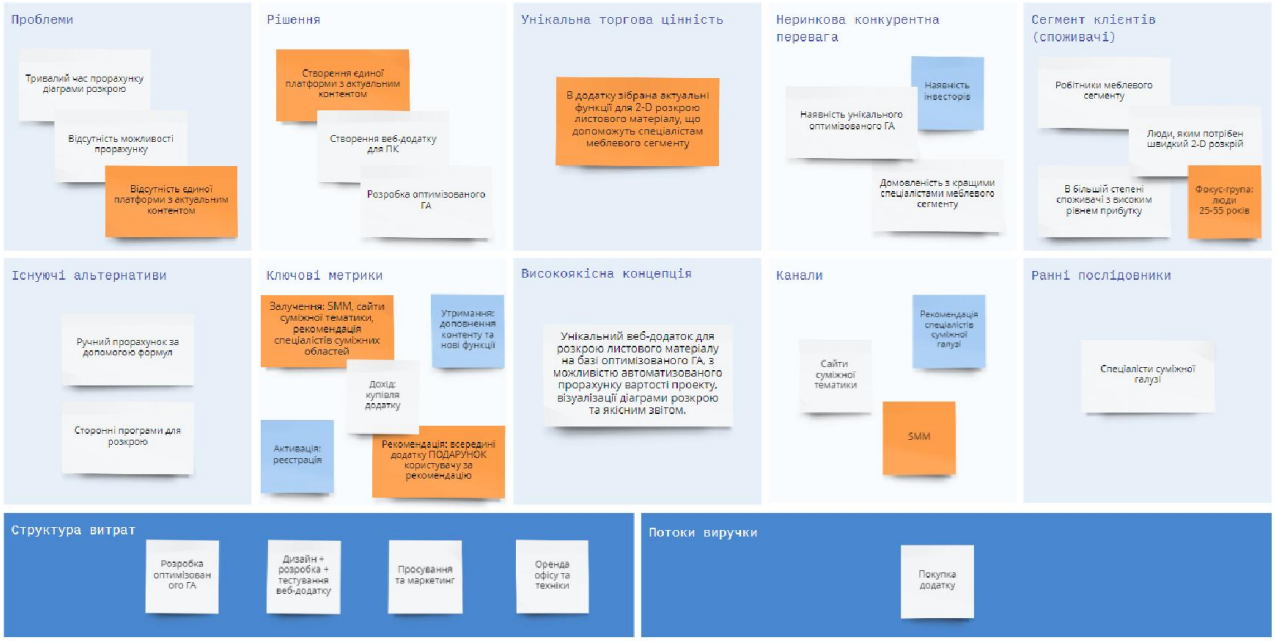


Рисунок 2.7 – LeanCanvas.

Воркшопи допомогли визначити те, що є важливим для продукту, а саме:

– сегмент клієнтів (споживачі);

– ранні послідовники;

– проблеми;

– існуючі альтернативи;

– унікальна торгова цінність;

– рішення;

– канали;

– потоки виручки;
– структура витрат;
– ключові метрики;
– неринкова конкурентна перевага;
– високоякісна концепція.

Наступним важливим етапом планування проекту є прорахунок часу на його виконання.Чіткий розрахунок часу, який витрачається на проект, є дуже важливим нюансом у діяльності кожного фрілансера. Це обумовлено тим, що часто ви продаєте замовникам свій особистий час і у разі неправильних розрахунків ви ризикуєте заробити в рази менше, ніж спочатку планували. Найчастіше правильна початкова оцінка є гарантією успіху проекту.

Відомі дві основні причини, які пояснюють усю важливість правильного розрахунку часових витрат:

– від розрахунку часу залежить встановлення термінів для надання результатів отриманого замовлення та планування своєї роботи. Таким чином, правильно розраховуючи час, ви можете зарекомендувати себе як хорошого спеціаліста та заслужити гарний авторитет;

– витрачений час є основним показником, що впливає на вартість послуг і як результат на рентабельність вашої діяльності.

Помилятися в розрахунках можуть не тільки новачки, а й досвідчені фрілансери, оскільки оцінка часу, що витрачається, досить складний момент, якому дуже важливо навчитися ще на початку своєї діяльності для досягнення поставлених цілей і успіху в цій сфері.Найчастіше фахівці недооцінюють обсяг роботи, яку беруться, отже, визначається неправильний розмір бюджету, необхідний для створення проекту. Неправильно розрахувавши обсяг витраченого часу, високі ризики заробити значно менше, ніж планували в розрахунку на годину.

За помилкового розрахунку часуможна не тільки втратити частину потенційного доходу, неправильно сформувавши бюджет проекту, але й зірвати

всі призначені терміни. Таким чином, є ризик підірвати свою репутацію і втратити роботодавця[65].

Для планування процесів та прорахунку часу було обраноJira[66] як основне середовище. Ввесь процес роботи над проектом поділено на 4 спринти:

– перший спринт має 13 завдань;
– другий спринт – 7 завдань;
– третій спринт має 14 завдань;
– четвертий спринт має 5 завдань.

Також для кожного із процесів визначено:

– priority;
– story points;
– time tracking;
– status.

Нарисунку 2.8зображено спринти проекту.

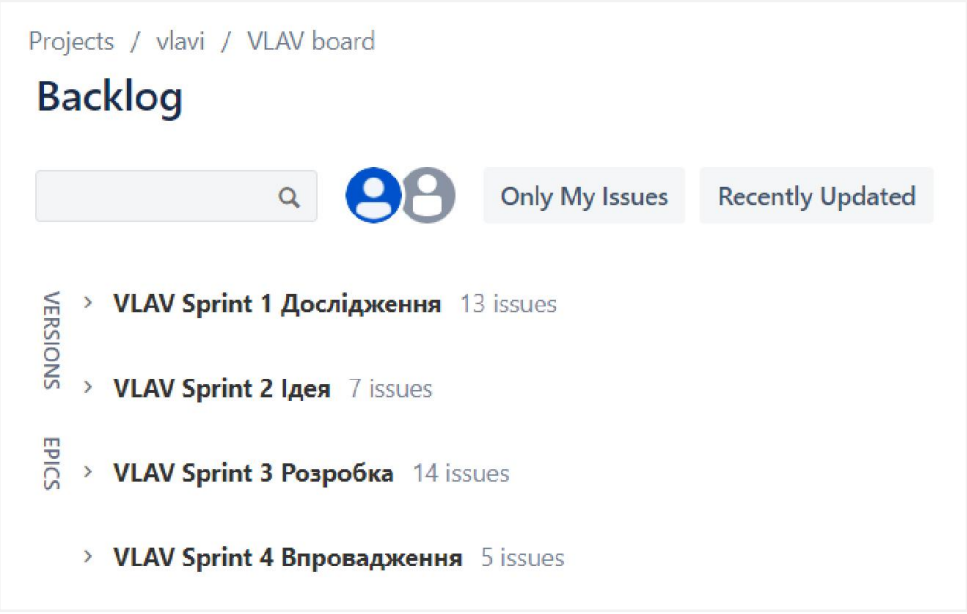
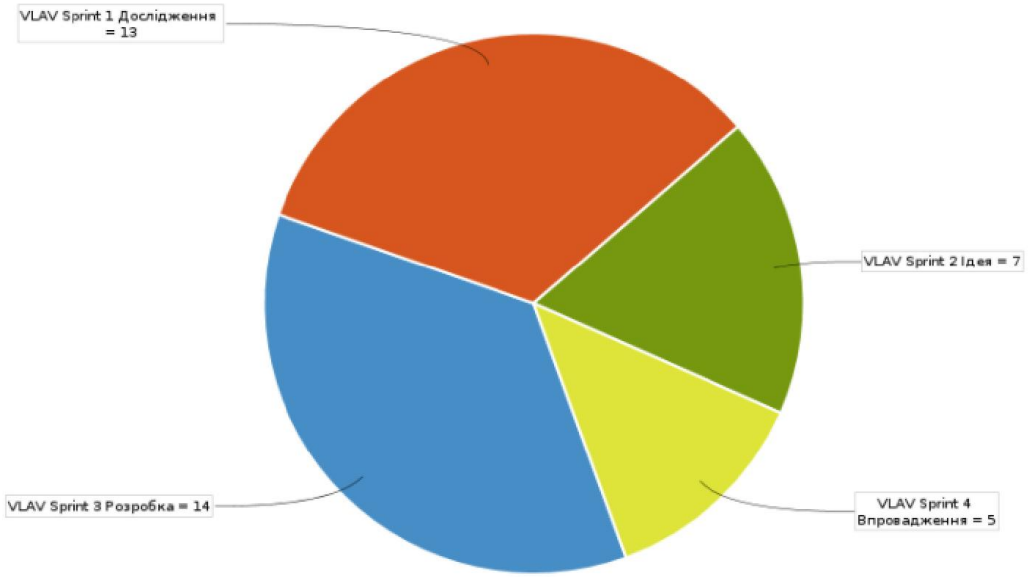


Рисунок 2.8 – Спринти проекту.

Нарисунку 2.9 – зображенаPieChartReport (Sprint). Дана діаграма розкриває відсоткове відношення кількостей задач по кожному спринту.



Data Table

	Issues	%
VLAV Sprint 3 Розробка	14	35%
VLAV Sprint 1 Дослідження	13	33%
VLAV Sprint 2 Ідея	7	17%
VLAV Sprint 4 Впровадження	5	12%

Рисунок 2.9 – Діаграма звітуPie Chart Report (Sprint) з додатком.

Як результат, бачимо:

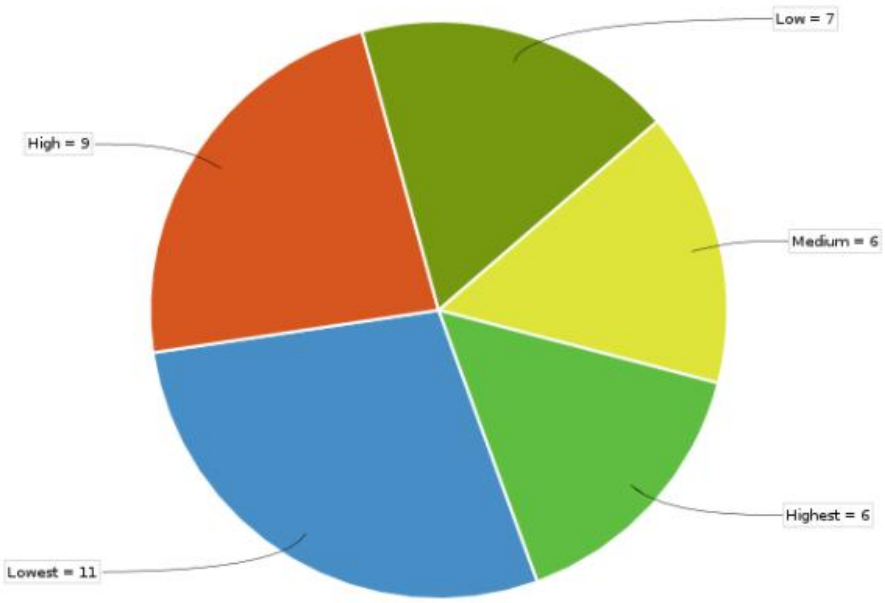
– спринт 3має найбільшу кількість процесів, складає 35%, тобто 14 задач;

– спринт 1 складає 33%, тобто 13 задач;

– спринт 2 складає 17%, тобто 7 задач;

– спринт 4 має найменшу кількість процесів, складає 12%, тобто 5 задач.

На рисунку 2.10 – зображенаPieChartReport (Priority) – діаграма розкриває відсоткове відношення пріоритетів кожної задачі проекту.



Data Table

	Issues	%
Lowest	11	28%
High	9	23%
Low	7	17%
Medium	6	15%
Highest	6	15%

Рисунок 2.10 – Діаграма звітуPie Chart Report (Priority) з додатком.

Як результат, бачимо:

– найбільше задач із пріоритетом «Lowest» – 11, що складає 28%;

– задач із пріоритетом «High» – 9, що складає 23%;

– задач із пріоритетом «Low» – 7, що складає 17%;

– найменше задач із пріоритетом «Medium» та «Highest» – 6, що складає по15%.

На рисунку 2.11 – зображеноTimeTrakingReport – розкриває початкові та поточні оцінки часу для задач у проекті.

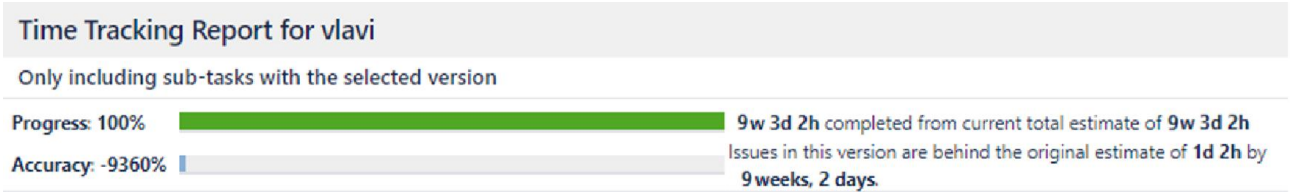


Рисунок 2.11 – Діаграма звіту «Time Traking Report».

Звіт відстеження часу відображає корисну інформацію щодо відстеження часу по завданням, пов’язаних із конкретною версією проекту. У цьому звіті наведено вихідні і поточні оцінки часу для всіх завдань, а також випередження або відставання від початкового графіка.

Звіт також містить дві гістограми (над таблицею), які представляють сукупну інформацію про час відстеження для версії:

– перша гістограма – Progress – показує відсоток завершених проблем (зелений) і незавершених проблем (помаранчевий);

– друга гістограма – Accuracy – показує відсоток виконання або відставання від графіка.

2.2 Функціонал та структураweb-додатку

Діаграма прецедентів в UML – це діаграма, що показує взаємозв'язок між учасниками системи та прецедентами. Він також перекладається як діаграма варіантів використання. Основна мета UML – графічно зобразитиархітектуру проекту[67].

У мові UML для формалізації функціональнихвимог застосовуються діаграми використання.

На діаграмі використання зображуються:

1. Актори –групи осіб або систем, що взаємодіють знашим додатком;

2. Варіанти використання (прецеденти) – сервіси,які наш додаток надає акторам;

3. Коментарі;

4. Відносини між елементами діаграми.

Види відносин:

– ставлення асоціації (associatio);

– ставлення розширення (extend relationship);

– ставлення включення (include relationship);

– ставлення узагальнення (generalization relationship).

Чи означає, що два ібільш актора можуть взаємодіяти з одним і тим жебезліччю прецедентів однаковим чином[67].

На рисунку 2.12 зображена реалізована в програмі Draw.io[68]діаграма варіантів використання.

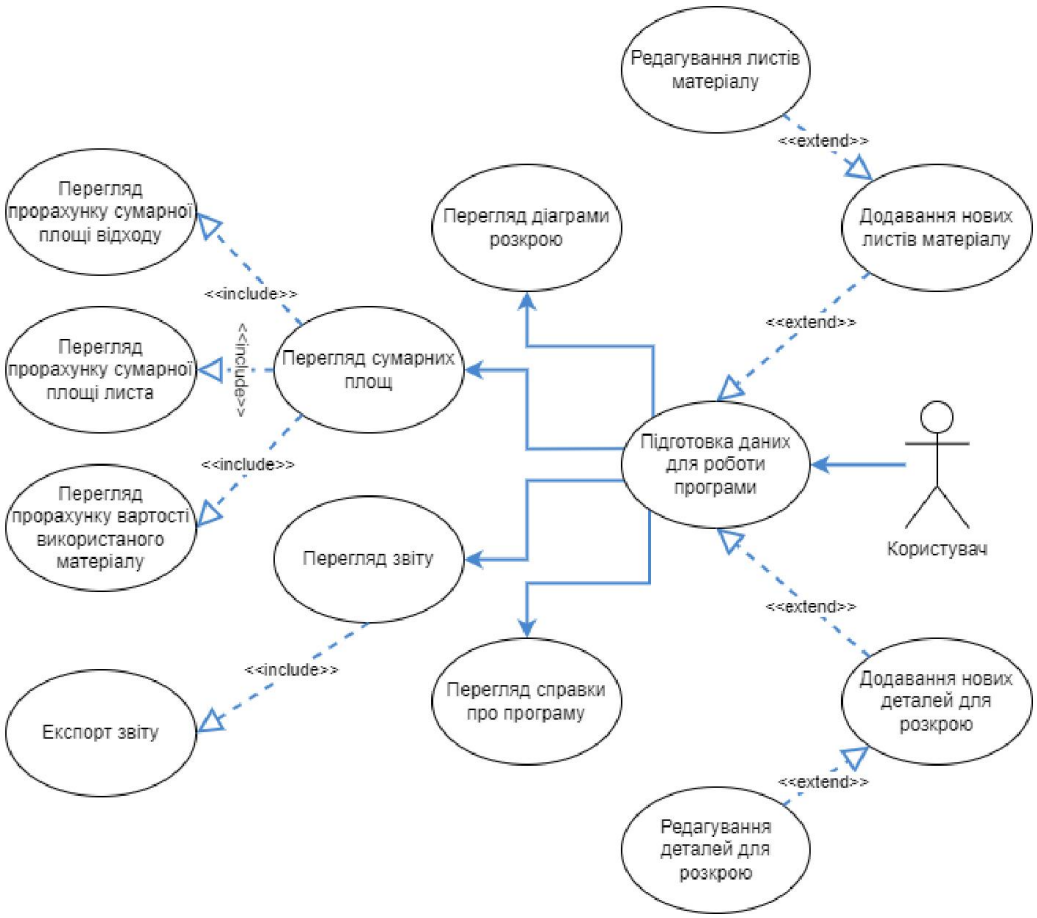


Рисунок2.12–Діаграмаваріантіввикористаннядлямайбутнього web-додатку.

Діюча особа (Actor) – це роль, що користувач відіграє стосовно додатку. Діючі особи являють собою ролі, а не конкретних людей або найменування робіт.

Актором для даного проекту є «Користувач».

До особливостей даної ролі відносять:

– є користувачем додатку;
– при необхідності може додати нові листи матеріалу;
– має змогу додавати та редагувати нові деталі для розкрою;
– може переглянути діаграму розкрою для створеного ним проекту;
– може переглянути прорахунок сумарної площі листа матеріалу, деталей, відходу, а також ціну за використаний матеріал;
– має змогу переглянути звіт про результати роботи з проектом;
– при необхідності може експортувати дані звіту.

Діаграми станів (Statechart Diagram) використовуються для опису поведінки складних систем. За допомогою них можна визначити всі можливі стани, до яких може бути наближена данасутність. Ці діаграми зазвичай використовуються для опису поведінки одного об'єкту в декількох прецедентах[69].

У мові UML станом називають період в житті користувача, протягом якого він задовольняє якусь умову, виконує певну діяльність або очікує деяку подію. Далі зображено діаграму станів для викладача. Оскільки він має змогу змінювати дані сайту, це відображено на діаграмі станів.

На рисунку 2.13 зображена діаграма станів для користувача.

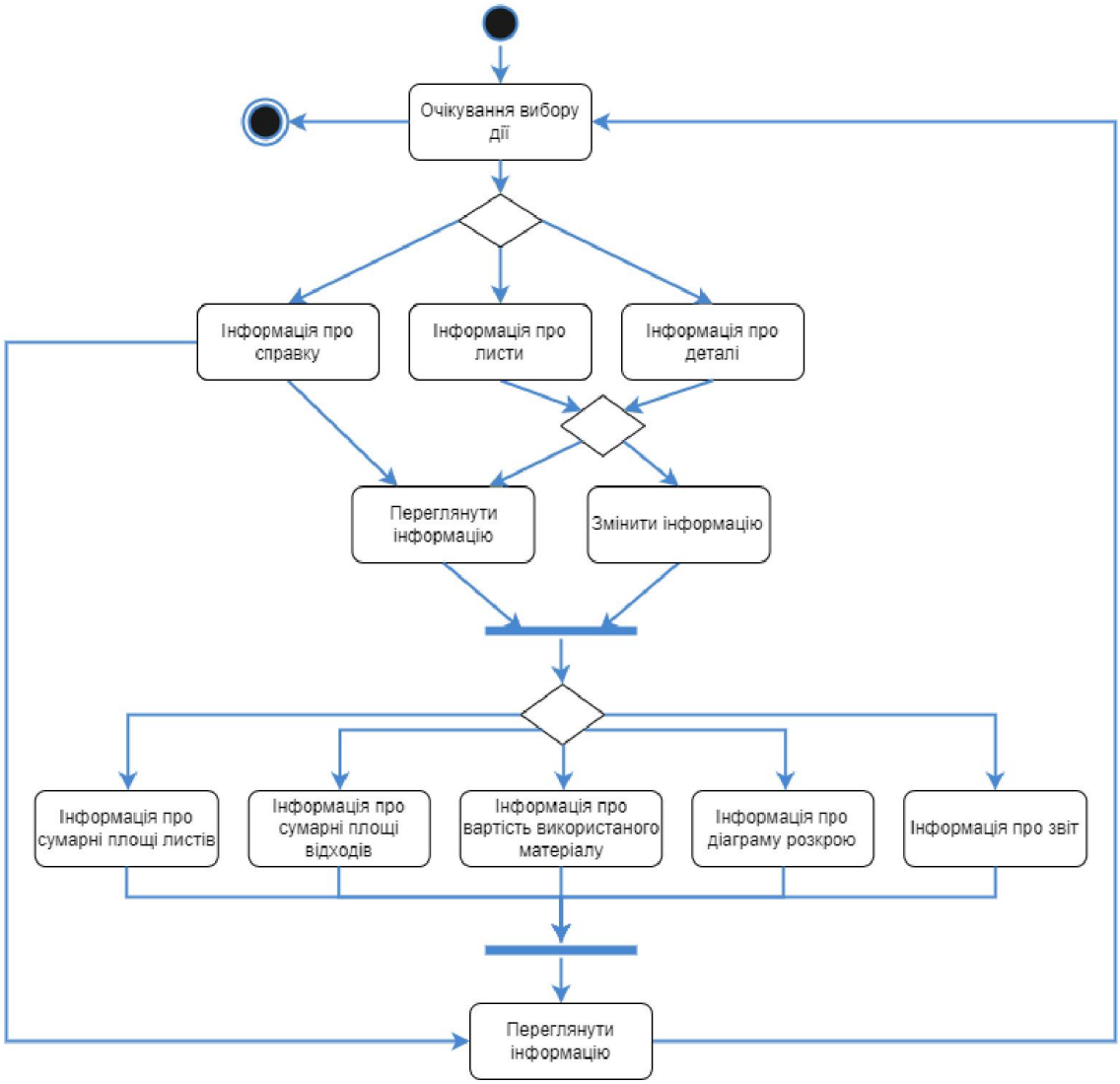


Рисунок2.13–Діаграмастанівдлякористувача

Фундамент сайту – правильно створена структура. При цьому немає відповіді на питання, якою вона має бути. Ясно тільки, що для кожного проекту підбирати її необхідно строго індивідуально. Також існують певні вимоги пошукових систем, яким потрібно слідувати в будь-якому випадку[70].

Правильна структура сайту – це система розташування сторінок сайту по чітко сформованої логічної схемою, структуру можна позначити, як ієрархію всіх сторінок сайту, їх приналежність до тих чи інших каталогах і папках.Структура сайту це шляху отримання користувачем інформації, яка запитується.

Існує кілька видів структур сайту:

1. Лінійна. Найпростіша структура – тут кожна сторінка посилається на іншу, одна за одною. Гості сторінкипроходить через весь сайт по одному шляху – від початку до кінця. Таку структуру мають презентації та односторінкові сайти.Їх завданням є прогулянка користувача по точному запланованому маршруту від початку до кінцясайту, і не датийому нікуди звернути;

2. Блокова. Така конструкція зустрічається дуже рідко, оскільки не підходить для всіх сайтів. В основному – для тих, хто представляє продукт. Кожен блок пояснює, у чому перевага, і всі сторінки пов'язані одна з одною. Тому просувати такий сайт ви практично готові праворуч від контенту;

3. Деревоподібна. Це найбільш поширена форма побудови сайту, і зрозуміло, чому вона дозволяє чітко визначати ієрархію сторінок, пов'язувати їх з панірувальними сухарями. Дана схема найкраще підходить для просування сайту, і саме на цьому має ґрунтуватися більша частина ресурсів в інтернеті [71].

Для проектування web-додаткубуло обрано деревовидну структуру, оскільки дана система має досить велику кількість сторінок і просте у розумінні та навігації меню.На рисунку 2.14 зображена структураweb-додатку на базі генетичного алгоритму для задач розкрою в умовах фіксованих 2-D обмежень.

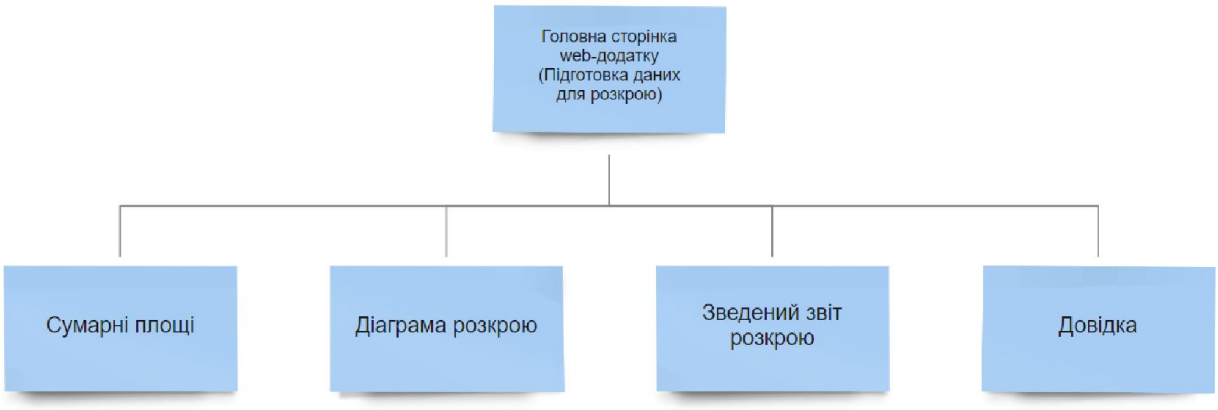


Рисунок 2.14 – Структура майбутнього web-додатку

2.3 Розробка логотипу та дизайну

За основу було взято загальну концепцію дизайну додатків для розкрою комплектуючих корпусних меблів. Були враховані дизайнерські рішення з проаналізованих додатків: зрозумілий вигляд меню, стриманий підбір шрифту, простий не відволікаючий від роботи з додатком дизайн переважно світлих чи сірих відтінків[72].

Для того, щоб досягнути кращої ефективності сприйняття матеріалу у додатку необхідно обрати правильну кольорову гамму.Оскільки додаток не має за ціль зацікавити чи привабити око користувача, а несе лише практичну цінність, Мною було обрано кілька відтінків сірого та білий кольори, що зображено на рисунку 2.15.

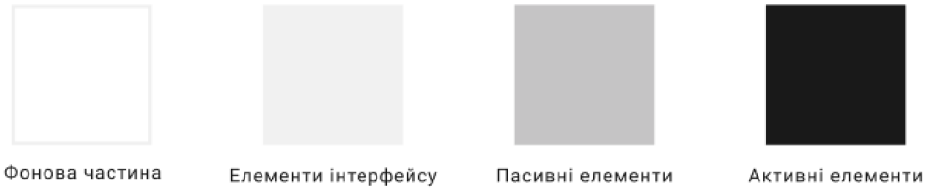


Рисунок 2.15–Обрані кольори

Для стилізації додатку було обрано переважно сірий колір різних відтінків. Нейтральність сірого не скасовує того емоційного ефекту, який він здатний. Сірий асоціюється з формальністю, меланхолією, нейтральністю, сумом, стриманістю, консерватизмом, мудрістю, серйозністю, байдужістю.

Сірий у різний його відтінках є дивовижною грою білого і чорного, що дає той чи інший цікавий ефект. [73].

Дане рішення використане в розробці логотипу та макету.

Для оформлення текстової частини було обрано шрифт «Roboto».

Я вирішила використати несистемний Roboto в інтерфейсі, тому що він

здався мені нейтральним і придатним для дизайну. Це дуже популярне в наш час шрифтове рішення, просте, що легко читається, і ідеально сумісне з будь-якими пристроями. Roboto має подвійну природу і це не ставить під загрозу його гротескні літери, змушуючи їх рухатися у незмінному ритмі, що дозволяє йому зберігати природну ширину. Шрифт Roboto близький до курсивного написання, але символи залишаються прямими[74].

На рисунку 2.16 зображено текстове рішення для реалізації додатку.

Рисунок 2.16–Шрифтове рішення додатку

Розробка логотипу – дуже важливий етап при створенні сайту чи додатку. Логотип грає велику, а іноді і ключову, роль у формуванні якісного дизайну веб проекту. Традиційно цей елемент веб дизайну розміщують в самій видною частини сайту (частіше в його шапці). Логотип беззастережно додає солідності і серйозності ресурсу. Також використовується при створенні фірмової іконки сайту для вікна браузера (фавікона) [75].

Для реалізації логотипу було обрано першу букву назви компанії замовника даного ресурсу«VlaVirchenko». Кольоровим рішенням є обраний раніше темно-сірий колір, котрий є акцентним в даній роботі. Мінімалістичний стиль який характеризується мінімальною кількістю кольорів, відсутністю тіней та об’єму гарно підійшов для даного лого.

На рисунку 2.17 зображений логотип для web-додатку на базі генетичного алгоритму для задач розкрою в умовах фіксованих 2-D обмежень.

Рисунок 2.17– Логотип для web-додатку.

Постійно видимі сторінки в меню стають все більш поширеними. Плиткові решітки з навігаційними розділами привертають більше уваги. Все що потрібно для навігації – це вашу увагу. Проектуйте навігацію свого сайту так, щоб відвідувачі могли легко і швидко розібратися в ній.Вертикальне меню не гірше, ніж горизонтальне. Довжина вертикального меню може бути майже нескінченною, а горизонтального – обмежена шириною вікна[76].

У більшості випадків при побіжному перегляді сторінок погляд користувачів рухається по F-образної траєкторії. Це так званий F-Shaped Pattern. Отже, найбільш важливу інформацію варто розміщувати зліва[77].

На рисунку 2.18 зображена траєкторія переглядів F-Shaped Pattern.

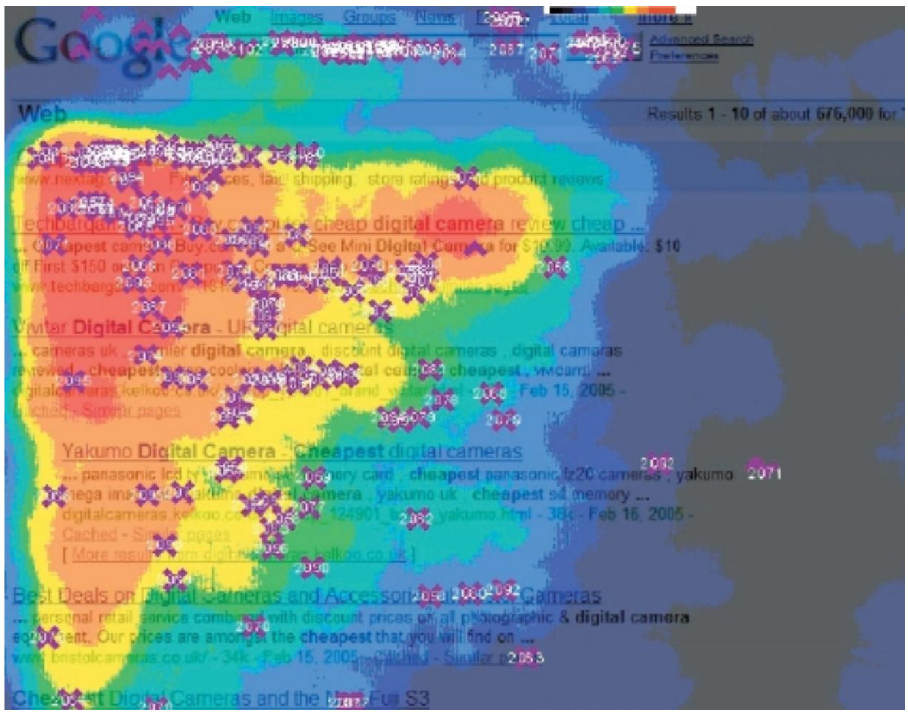


Рисунок 2.18 – Траєкторія переглядів F-Shaped Pattern.

Для навігації в системі було обрано бокове меню, адже в додаток буде наповнено новими функціями та можливостями.

Відповідно до розробленої структури, в програмі Figma[78] було спроектовано макети 4 сторінок додатку:

– підготовка даних для розкрою;
– сумарні площі та ціна;
– діаграма розкрою;
– звіт.

На рисунку 2.19 зображений макет всіх сторінок для web-додатку на базі генетичного алгоритму для задач розкрою в умовах фіксованих 2-D обмежень.

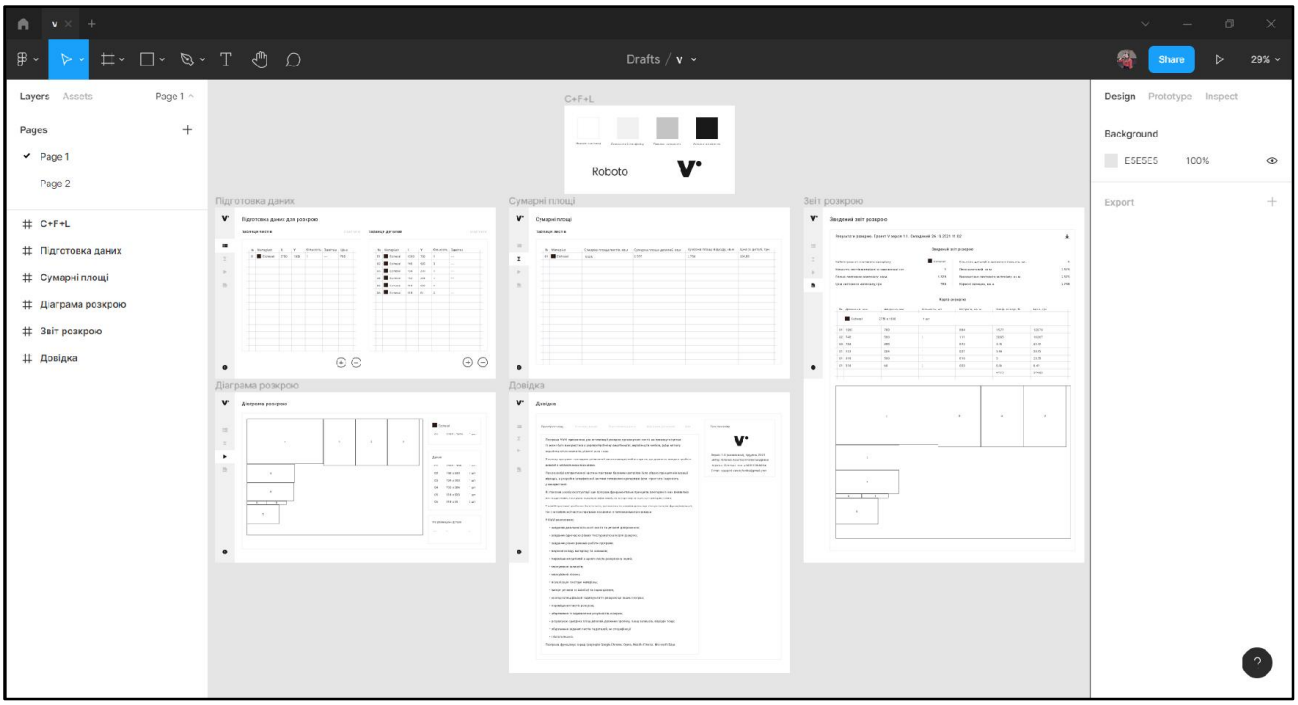


Рисунок 2.19 – Макет для web-додатку

РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА РОЗРОБКИ WEB-ДОДАТКУ НА БАЗІ ГЕНЕТИЧНОГО АЛГОРИТМУ В УМОВАХ ФІКСОВАНИХ 2-DO ОБМЕЖЕНЬ

3.1 Вибір та обґрунтування веб-технологій для розробки програмного забезпечення

3.1.1. Вибір та обґрунтування використання web-сервісу для розробки додатку. Веб-сервіс має безліч принад в зрівнянні з "настільними" програмами. Насамперед веб-сервіс не потребує встановлення на комп'ютер. Для доступу до нього достатньо підключення до Інтернету.

На відміну від "настільних" програм, веб-сервіс може працювати під будь-якою операційною системою (Windows, MacOS, Linux). Все, що потрібно – встановлений браузер останніх версій (Firefox, Google Chrome, Opera, Internet Explorer).

Основа даних, розроблена в онлайн-системі, зберігається на сервері в спеціалізованому дата-центрі. Завдяки запасному копіюванню, вона надійно захищена від втрат, а зашифрований канал передачі гарантує їх конфіденційність і захист від перехоплення.

Веб-сервер обслуговується фахівцями, доступ до нього чітко регламентований. Вам не потрібно більше пахнути витрати, пов'язані з запасним копіюванням даних, сервісом і налаштуванням обладнання. Ще не треба спостерігати за оновленням програми – все відбувається автоматично.

Веб-сервіс менш вимогливий до ресурсів комп'ютера, ніж "настільна" програма, оскільки всі складні обчислення відбуваються на стороні сервера.

Єдина невід'ємна умова для роботи з онлайн-системою – наявність постійного широкосмугового включення до Онлайн.

На мою думку вагомими аргументами даної системи є наступні положення:

1. Програми найчастіше є платними. Не кожен погодиться купувати ліцензійну версію Photoshop[79], щоб створити макет візитки;

2. Завантаження «лівих» програм з Інтернету загрожує небезпекою підхопити вірус, який може наробити багато пакостей у вашому комп'ютері;

3. Програми посідають місце на жорсткому диску. Коли воно закінчується, це неприємно[80].

3.1.2 Вибір та обґрунтування використаннямови HTML. HTML – це абревіатура від мови розмітки гіпертексту. Визнана у всьому світі, мова програмування, яка використовується для форматування веб-сторінок. Програмісти використовують її в поєднанні з каскадними таблицями стилів і JavaScript, щоб надати веб-сторінкам бажаний вигляд і відчуття. Незважаючи на наявність інших альтернатив, це вибір для створення веб-сторінок. На HTML-сторінці є два основних розділи: head і body. Дані, які описують сторінку, яку також називають метаданими, розташовані всередині head, тоді як body містить всі теги, які необхідні для представлення видимого вмісту сформованої сторінки. HTML є мовою, незалежною від платформи, тому її можна використовувати в будь-яку платформу, як-от Windows, Linux, Macintosh тощо [81].

Перевагами HTML є:

- широко використовується;
- кожен браузер підтримує мову HTML;
- легко навчатися та використовувати;
- має невелику вагу і швидко завантажується;
- не потрібно купувати будь-яке додаткове програмне забезпечення, оскільки воно за замовчуванням є в кожному вікні;
- простий у використанні;
- вільний синтаксис;
- досить легко писати;

– що його легко використовувати навіть початківцям програмістам;

– дозволяє використовувати шаблони, що полегшує розробку веб-сторінки;

– дуже корисно для початківців у сфері веб-дизайну;

– створений майже на кожному веб-сайті, якщо не на всіх веб-сайтах;

– все частіше використовується для зберігання даних як синтаксис XML;

– безкоштовна – вам не потрібно купувати програмне забезпечення;

– за замовчуванням присутній у кожному вікні, тому вам не потрібно купувати програмне забезпечення, яке коштує занадто дорого;

– має багато тегів і атрибутів, які можуть скоротити ваш рядок коду.

3.1.3 Вибір та обґрунтування використаннямови JavaScript для розроблення програмного забезпечення. JavaScript – це мова програмування, що є прототипно-орієнтованим. Він використовується багатьма фахівцями, які цінують його за гнучкість, підтримку більшістю браузерів, високу швидкість роботи і інші переваги.

Одне з ключових переваг даного програмного продукту – це підтримка практично всіма відомими і найпопулярнішими браузерами. Виділимо інші плюси JavaScript:

– зрозумілий для користувачів, які не є професійними програмістами;

– пряме підключення скриптів до HTML коду;

– можливість запуску програм в браузері і на сервері;

– широкий вибір корисних функціональних параметрів.

Мова програмування JavaScript постійно вдосконалюється, розробник випускає оновлення, в які додаються нові функції. Оновлені версії враховують недоробки минулих продуктів та побажання користувачів. З додатком можна взаємодіяти через звичайні текстові редактори, включаючи продукти Microsoft Office. Навчитися працювати з JavaScript можна самостійно, використовуючи

спеціальні навчальні посібники (їх можна знайти в Мережі), практикуючись і радячись з програмістами[82].

Причинами вибору мови JavaScript для програмної реалізації додатку є:

– повна інтеграція з html та css;

– підтримка всіма розповсюдженими браузерами.

3.1.4 Вибір та обґрунтування використанняTypeScript. Він представляє мову програмування на основі JavaScript:

– мова назад сумісний з JavaScript – якщо ви згодувати компілятору чистий JavaScript, компілятор виплюне вам ваш же JS, і не скаже що це помилка. Можна писати змішаний код (наприклад, модулі / методи використовуючи синтаксис TypeScript, а реалізацію методів без типізації на чистому JS) – це теж буде валідності;

– мова розроблений компанією Microsoft;

– система для роботи з модулями / класами – можна створити інтерфейси, модулі, класи;

– можна наслідувати інтерфейси (в тому числі множинне успадкування), класи;

– можна описувати власні типи даних;

– можна створювати універсальніінтерфейси (generic interfaces);

– можна описати тип змінної (або властивостей об'єкта), або описати яким інтерфейсом повинен володіти об'єкт на який посилається змінна[83].

3.1.5 Вибір та обґрунтування використанняVisual Studio Code. Це найпопулярніша безкоштовна програма для розробників, яка допомагає писати код. Наприклад:

– знає синтаксис різних мов програмування та допомагає вам не помилитися у точці з комою або дужкою;

– сама підставляє деякі поширені фрагменти коду;

– пам'ятає назви ваших змінних і нагадує їх, щоб не було помилок;

– вміє завантажувати код на Гіт;

– допомагає налагодити код;

– підтримує плагіни, які перетворюють її на мегакомбайн для розробника.

VSCode – один з найпопулярніших редакторів коду, тому що він безкоштовний і відкритий, його можна зробити будь-яким під свої завдання[84].

3.2 Проектування генетичного алгоритму для роботи додатку

Генетичні алгоритми (ГА) (Genetic Algorithms) поєднують у собі елементи детерміністичного і стохастичного підходів, тому часто відносять до групи адаптивних методів на вирішення завдань пошуку та оптимізації. ГА успішно застосовуються у поєднанні з аналітичними методами оптимізації або іншими алгоритмами випадкового пошуку. Принцип роботи генетичних алгоритмів заснований на моделюванні запозичених у природи процесів розвитку популяції біологічної особини генетично: перебудова набору хромосом генотипу з успадкуванням ділянок хромосомних наборів батьків (кросинговер), випадкова зміна генотипу (мутація). Процедура природного відбору, також запозичена у природи, спрямовано вдосконалення при зміні поколінь пристосованості членів популяції до «виживання», завдяки відбору особин з певними ознаками.

На рисунку 3.1 представлена блок-схема роботи генетичного алгоритму.

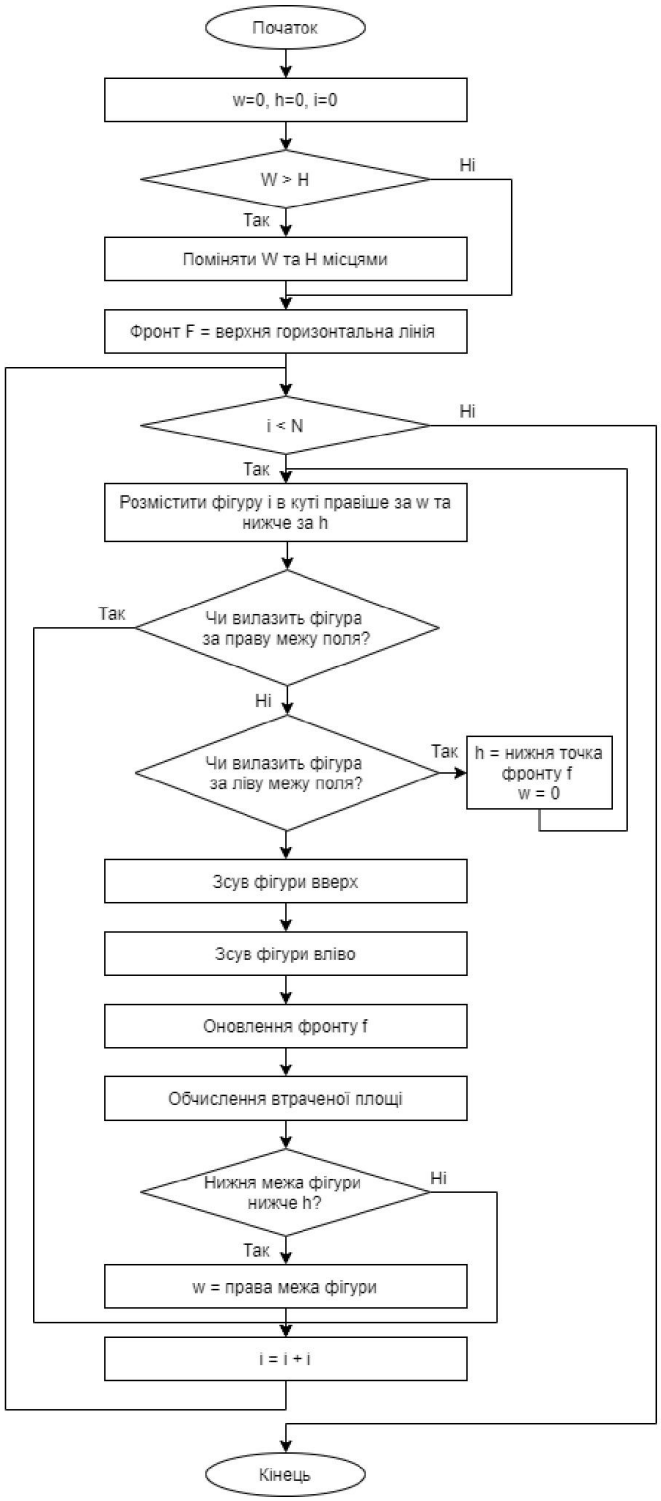


Рисунок 3.1 – Блок-схема роботи генетичного алгоритму.

Далі представлено опис алгоритму:

1. Початок роботи алгоритму.
2. Отримання вхідних даних.

Отримується масив лекал, які необхідно розмістити, і масив контурів, в яких розміщуютьсялекала. Дані передаються з інтерактивної підсистеми.

3. Розраховуємо, які лекала потрапляють до певних контурів.

Аналізуємо, які лекала потрапляють до певних контурів і складаємо список. Якщо в певнийконтур неможливо розмістити жодного лекала – цей контур автоматично потрапляє у відходи, і мийого видаляємо з масиву контурів.

4. Для кожного генома проводимо ітерацію.

Для всіх можливих варіантів наборів критеріїв (генів) проводимо ітерацію (розташовуємолекала).

5. Розраховуємо площу відходів і вибираємо m% найкращих результатів.

Для кожного можливого розташування лекала обчислюємо площу контурів, в які вже непотрапить жодне лекало, тобто ті контури, які підуть у відходи. Після цього з отриманихрезультатів вибираємо m% результатів (селекціонуємо найкращі гени), де площа відходівнайменша, враховуючи результати, які наближаються до тих m%.

6. Розраховуємо, які лекала потрапляють до нових контурів.

Аналізуємо, які лекала потрапляють до нових контурів і складаємо новий список лекал.

7. Чи є для утворених нових контурів лекала?

Якщо в нових контурах не можливо розмістити жодного лекала, переходим до кроку 8, віншому випадку переходимо до кроку 9.

8. Утворені контури відносимо до відходів.

Контури потрапляють до відходів і ми їх видаляємо з масиву контурів.

9. Чи є ще лекала для розміщення?

Перевіряємо, чи є у вихідному масив ще не розміщені лекала. Якщо немає, переходимо докроку 11. В іншому випадку переходимо до кроку 10.

10. Схрещуємо найкращі результати і отримуємо нові можливі геноми.

За допомогою генетичних операторів (кросинговеру, мутації та інших) схрещуємо отриманіпокоління для можливого зародження досконалішого гена. Надалідля виконання генетичнихоперацій не використовуємо тих критеріїв, які

дали негативні результати. Отримуємо нові можливігеноми (нову популяцію). Переходимо до кроку 4.

11. Виведення результатів.

На цьому кроці виводимо такі результати:

– послідовність розташування лекал у контурах з координатами розміщення лекал;

– процент відходів;

– які лекала не розміщені (якщо такі є).

12. Кінець роботи алгоритму.

Роботу даного алгоритму було представлено в тезахдо73rd Scientific Conference of Professors, Teachers, Researchers, Postgraduate Students and Students of the National University «Yuri Kondratyuk Poltava Polytechnic»[85].

З лістингом коду реалізації алгоритму можна ознайомитись у Додатку Г.

3.3 Розробка інтерфейсу додатку

Відповідно до розробленої структури та макету, була розроблено екрани додатку. Він містить всі основні структурні елементи, перехід за якими здійснюється за допомогою гіперпосилань.

Розглянемо детально кожну сторінку.

При переході за посиланням на додаток користувача зустрічатиме екран підготовки даних для розкрою. Таке рішення було прийнято, щоб користувач міг негайно приступити до роботи не чекаючи завантаження сторонніх анімацій. Користувачу представлено 2 таблиці:

– таблиця листів;

– таблиця деталей.

Для подальшої роботи додатку користувач в першу чергу має заповнити дані таблиці листів:

– матеріал;

– довжина листа;
– ширина листа;
– кількість;
– замітка (для більш зручного перегляду діаграми розкрою);
– ціна листа матеріалу.

Після чого можна заповнити таблицю деталей:

– матеріал листа (зазначений в таблиці листів);
– довжина листа;
– ширина листа;
– кількість;
– замітка (для більш зручного перегляду діаграми розкрою).

Також користувач може очистити таблиці при необхідності. На рисунку 3.2 зображено екран «Підготовка даних для розкрою».

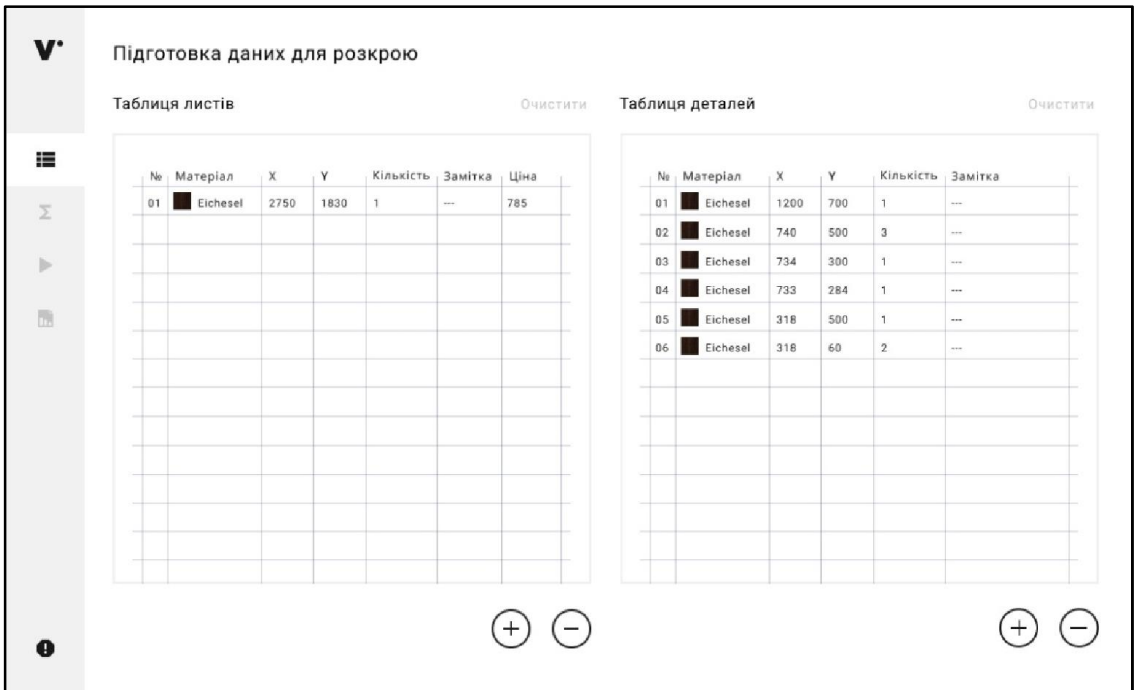


Рисунок 3.2 – Підготовка даних для розкрою.

Після коректного заповнення попередньої таблиці користувач може переглянути статистичні дані на сторінці «Сумарні площі»:

– матеріал;

– сумарна площа листів;
– сумарна площа деталей;
– сумарна площа відходу;
– ціна за деталі.

На рисунку3.3 зображений вікно зі статистичними даними.

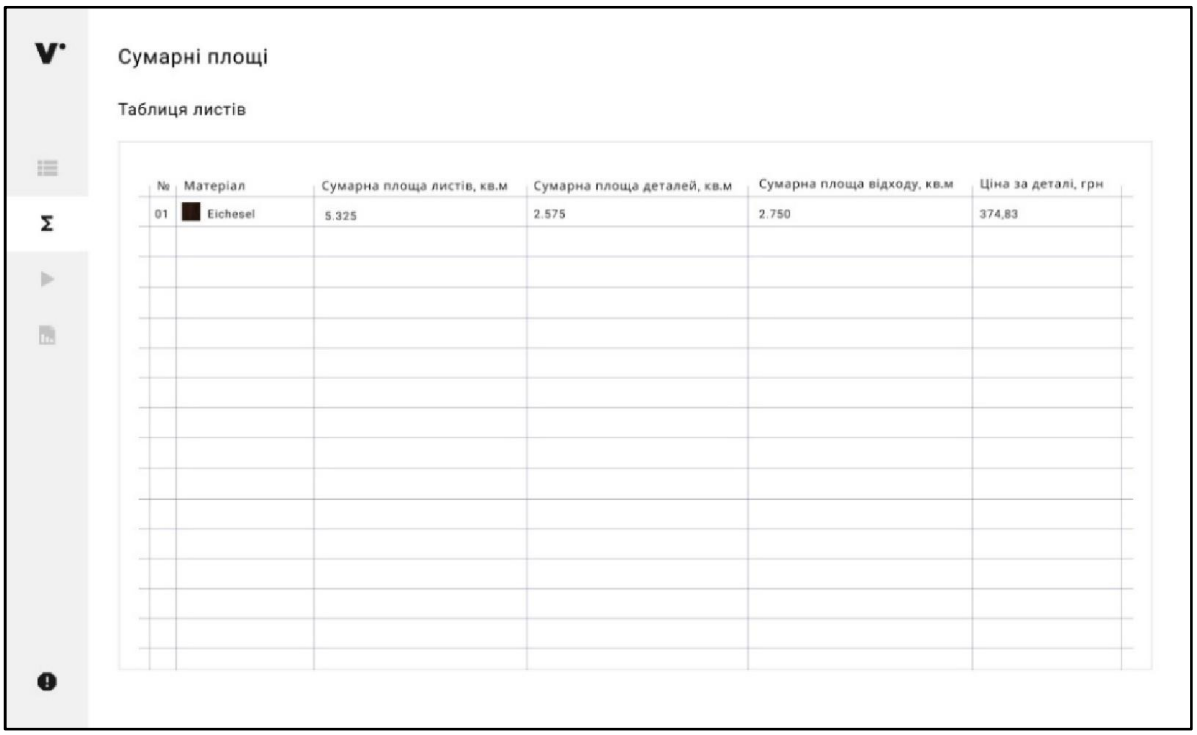


Рисунок 3.3 – Сумарні площі.

При переході до третього пункту меню можна ознайомитися з картою розкрою листового матеріалу на базі генетичного алгоритму. Також доступна інформація стосовно самого листа матеріалу, розміщених деталей та деталей, що не буди розміщені.На рисунку3.4 зображений екран перегляду діаграми розкрою.

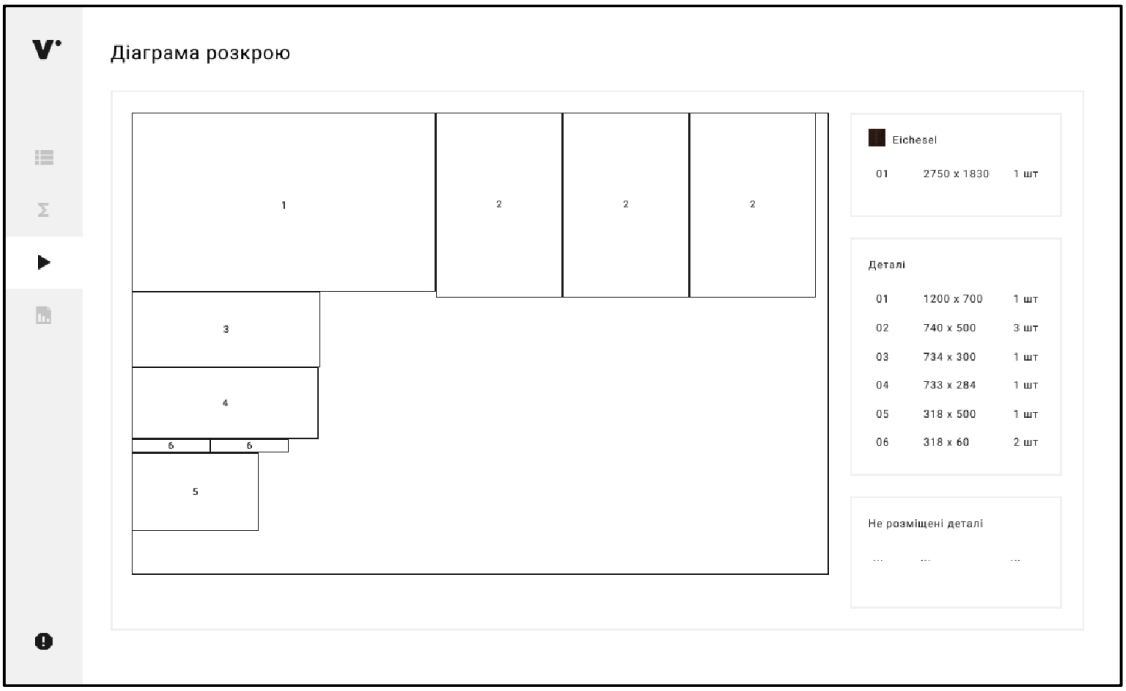


Рисунок 3.4 – Діаграма розкрою.

Якщо перейти до четвертого пункту меню, можна переглянути зведений звіт стосовно проекту з датою та часом користування додатком. Для ознайомлення виведено зведений звіт розкрою:

- найменування листового матеріалу;
- кількість листів матеріалу в замовленні, шт.;
- площа листового матеріалу, кв.м.;
- ціна листового матеріалу, грн;
- кількість деталей в замовленні всього, шт.;
- площа деталей, кв.м.;
- використано листового матеріалу, кв.м.;
- корисні залишки, кв.м..

Для зручного перегляду інформації стосовно деталей було виведено карту розкрою, а саме:

- довжина, мм.;
- ширина, мм.;
- кількість, шт.;

– витрата, кв. м.;

– коефіцієнт використання, %;

– ціна, грн..

Також було виведено окремо коефіцієнт використання для всіх деталей в цілому та повну ціну використаного матеріалу. Є можливість повторного перегляду діаграми розкрою листового матеріалу

При необхідності користувач може експортувати сформований звіт на свій девайс.

На рисунку 3.5 зображений екран зі сформованим звітом.

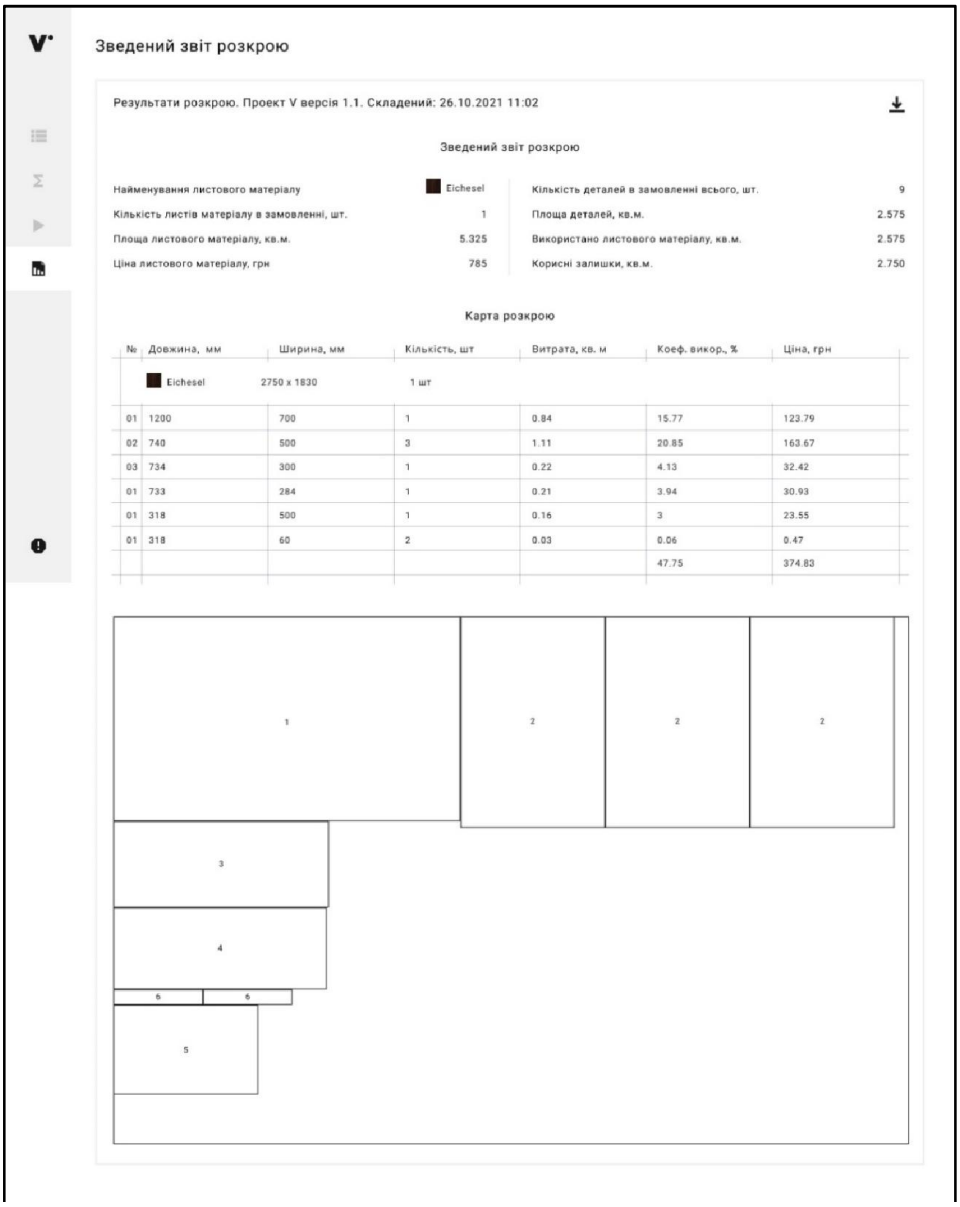


Рисунок 3.5 – Зведений звіт розкрою.

Якщо користувачу виявився незрозумілим якийсь із процесів роботи з додатком, він може скористатися пунктом меню «Довідки», та переглянути особливості експлуатації

- про програму;
- підготовка даних;
- сумарні площі;
- діаграма розкрою;
- звіт.

Також користувач може ознайомитися з відомостями про сам додаток:

- версія додатку;
- дата останнього оновлення;
- автор додатку;
- країна та місто створення додатку;
- контакти автора додатку.

На рисунку 3.6 зображений екран з особливостями експлуатації.

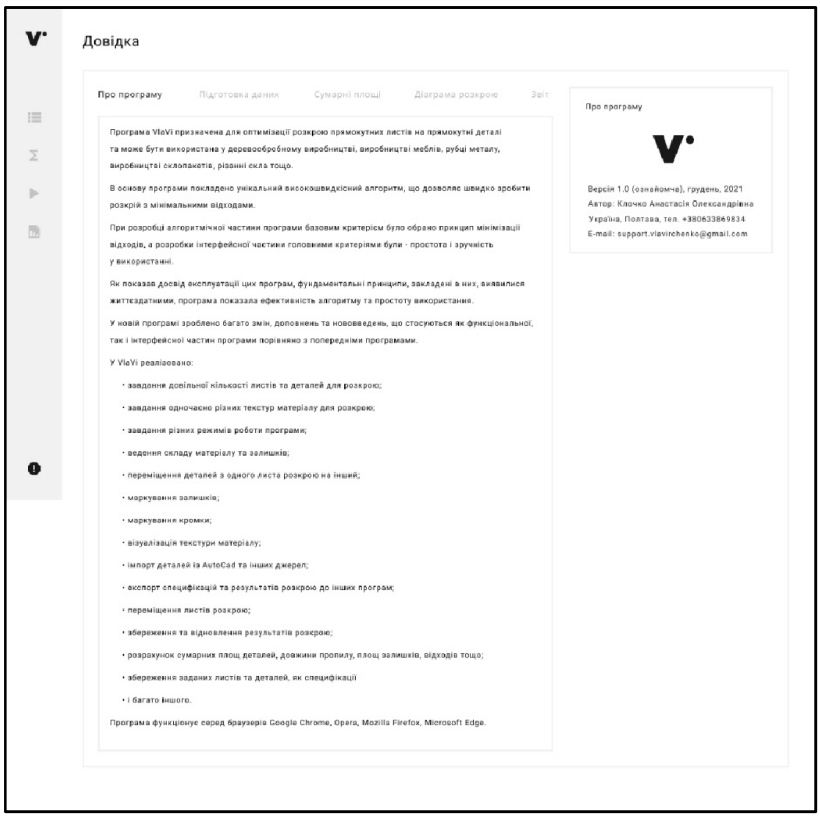


Рисунок 3.6 – Екран «Довідка».

РОЗДІЛ 4
ТЕСТУВАННЯ ТА ВВЕДЕННЯ В ЕКСПЛУАТАЦІЮ

4.1 Тестування ефективності роботи використаного ГА

Генетичні алгоритми належать до глобальних емпіричних методів оптимізації. Вони завдячують своєю появою Чарльзу Дарвіну та її еволюційної теорії, опублікованої 1859 р. у роботі «Походження видів», у якій проголошено основні принципи еволюційної теорії:

- спадковість;
- мінливість;
- природний відбір.

Засновником сучасної теорії генетичних алгоритмів вважається Джон Холланд (JohnHolland), який зумів застосувати теорію Дарвіна до штучних систем. Його робота «Адаптація в природних і штучних системах» («AdaptationinNaturalandArtificialSystems»), опублікована в 1975 р., стала класикою в цій галузі. У ній Холланд вперше запровадив термін генетичний алгоритм. Нині описаний там алгоритм називають класичним чи канонічним. Саме поняття генетичний алгоритм стало дуже широким і найчастіше до нього відносять алгоритми, що сильно відрізняються від класичного [86].

Ефективність роботи ГА прийнято оцінювати кількістю обчислень цільової функції. Чим менше тим краще. Після деяких функцій наведено результати роботи мого ГА (назвемо його QGA). Відразу зазначу, що результати цільової функції менше 0.001 теж зараховувалися як знайдений глобальний мінімум. Ось характеристики QGA:

- геном з фіксованим розміром;
- фіксована розрядність популяції;
- число генів рівне числу точок розриву кросовера;
- для схрещення відбирається 50% популяції;
- можливість мутації 95%;

- два найкращих індивіда;
- знищення однакових особин;
- чіткість оцінки: 0.001;
- результат розрахований за 50 циклами ГА.

Розуміючи, що ГА користуються стохастичністю, можна зробити висновок, щодля визначення її ефективності, варто зробити декілька запусків на деякій функції і лише згодом аналізувати результат.

Оцінка якості запропонованого алгоритму та приватних генетичних операторів проводилася за декількома напрямками: по-перше, порівняння ефективності пошуку в залежності від обраної символьної моделі, у тому числі від довжини хромосомного набору, по-друге, залежність покращення значення пристосувань особин популяції від параметрів генетичного алгоритму, насамперед методу вибору батьківських пар та реалізації механізму відбору.

Оцінка запропонованого алгоритму проводилася на низці тестових функцій. Це насамперед кілька тестових функцій De Jong'a, які часто використовуються для перевірки ефективності нових генетичних алгоритмів, крім того, були використані складні багатоекстремальні функції високої розмірності, які практично не вирішуються класичними методами[87].

Таблиця 4.1 – Тестові задачі для тестування генетичного алгоритму

j	**Тестова задача**	**Розмірність**	**Властивості**
1	De Jong 2	2	Овражна функція, один глобальний екстремум
2	De Jong 3	5	Розриви типу "скачок", максимум досягається на гіперкубі
3	De Jong 5	2	Один глобальний на гіперкубі, 24 локальні максимуми.
4	Griewank	2	Один глобальний та безліч локальних максимумів.

Далі представлено результати тестування генетичного алгоритму.

На рисунку4.1 зображеновідношення знайденого максимального

значення до реального максимуму за різної довжини кодування гена для 4 тестових функцій.

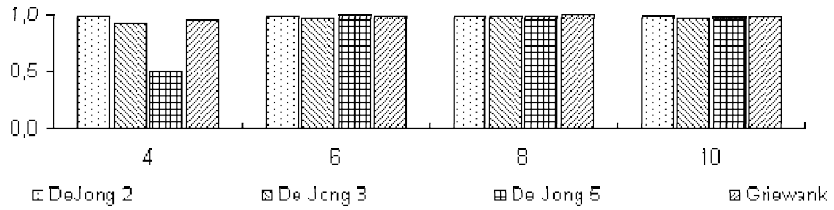


Рисунок 4.1 – Відношення знайденого максимального значення до реального максимуму.

На рисунку 4.2представлені графіки зміни пристосованості кращої особини у популяції при різній довжині кодування; безперервна лінія – L = 10, розривна – L=6, пунктирна – L=4.

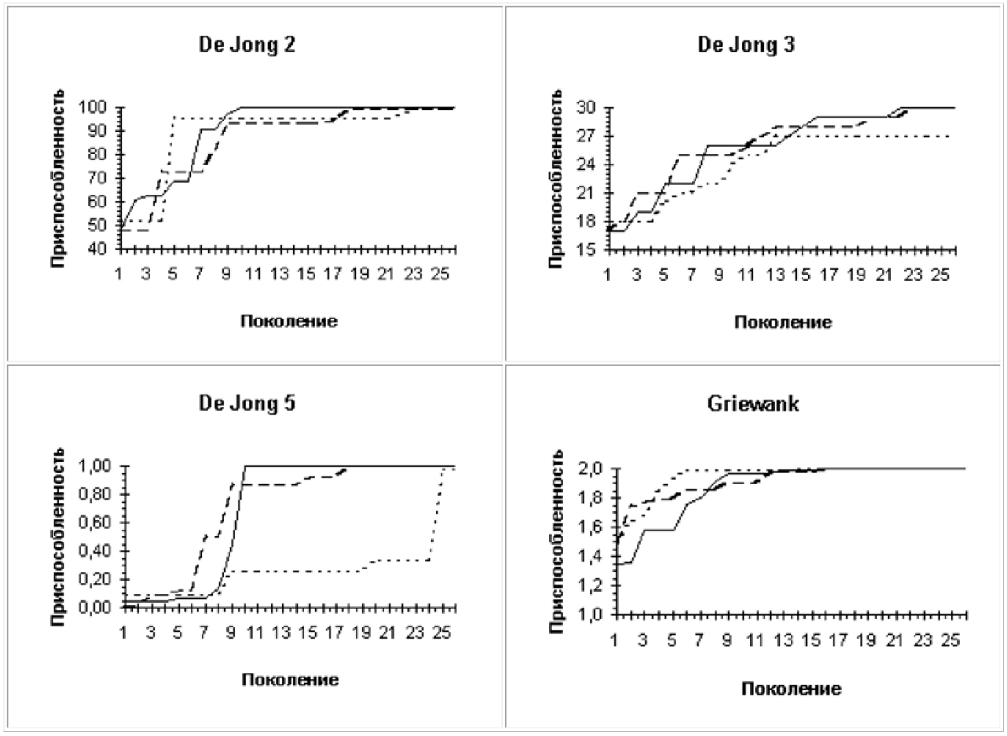


Рисунок 4.2 – Графіки пристосованості функцій.

На рисунку 4.3представлені зміна кращої та гіршої пристосувань особин у популяції. Неважко бачити, що збільшення довжини хромосомного набору не

так прискорює процес пошуку найкращого значення пристосованості, скільки впливає на поліпшення пристосувань найгірших особин у популяції, що призводить до більш швидкої збіжності алгоритму на шкоду вивченню простору пошуку.

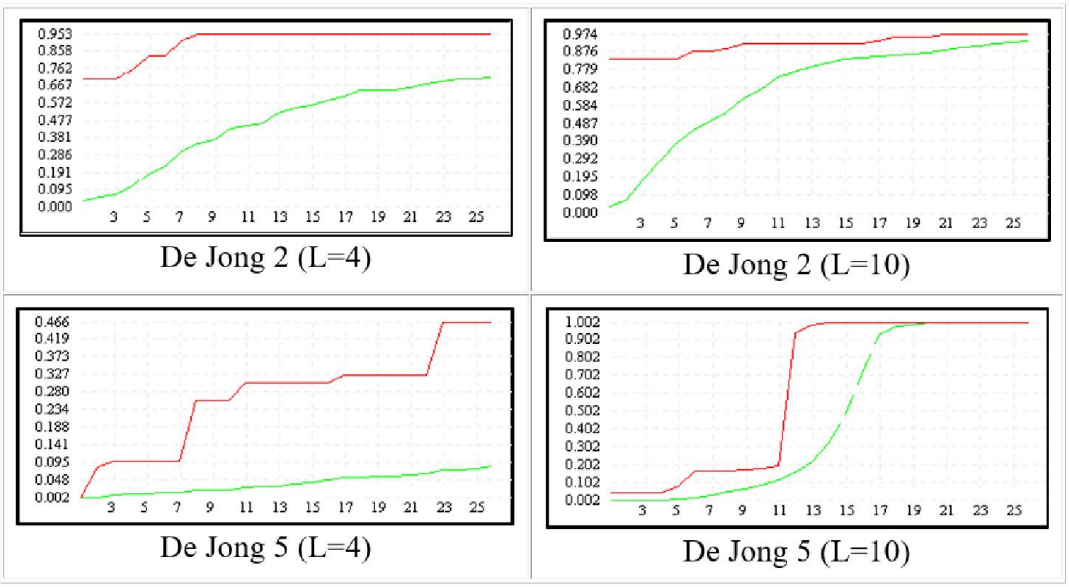


Рисунок 4.3 – Зміни пристосованості функцій.

Ефективність ГА при використанні різних операторів кросовера. На рисунку 4.4представлені діаграми з результатами, усереднені по 50 запускам (кожен запуск – близько 5000 обчислень для 10-мірних функцій чи 1000 інших).

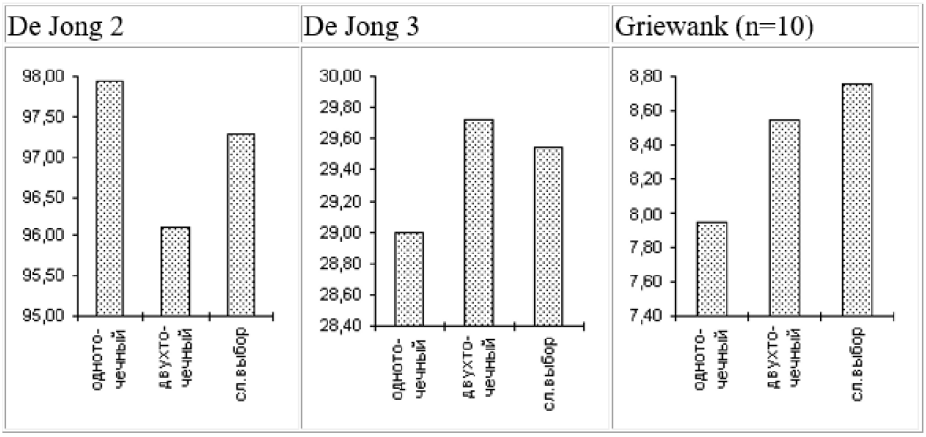


Рисунок 4.4 – Використання різних операторів кросовера для функцій.

4.2 Вибір методу тестуваннядодатку

Тестування грає життєву роль в процесі розробки і створення якісного програмного забезпечення. Необхідно серйозно ставитися до аналізу і проектування структурованого процесу, який забезпечить своєчасний і успішний випуск проекту[88].

У наші дні все більше і більше веб-додатків розробляються. І з кожним написаним рядком коду потенціал помилок виникає.

Взагалі кажучи, витрати на виправлення помилок експоненціально зростають, чим пізніше ви їх знайдете.

Етапи тестування веб-проектів:

1. Функціональне тестування. Перший крок веб-тестування забезпечує тестування функцій додатку. У Вікіпедії функціональне тестування описано так:

Функціональне тестування – це процес забезпечення якості (QA) і тип тестування в чорному ящику, який базує свої тестові випадки на специфікаціях програмного компонента, що перевіряється. Функції перевіряються, подаючи їх на вхід та вивчаючи вихід, а внутрішня структура програми розглядається рідко (на відміну від тестування з білого поля).

Функціональне тестування відбувається у вихідному коді, де додаток тестується на функціональні вимоги та технічні характеристики.

Зазвичай функціональне тестування включає:

- ідентифікація функцій, які повинно виконувати програмне забезпечення;
- введення та введення даних;
- виконання тестової справи;
- аналіз фактичних результатів.

Під час функціонального тестування моделюється фактичне використання додатку. Ідея полягає в тому, щоб максимально наблизитись до

реального використання додатку та створити тестові умови, пов'язані з вимогами користувачів[88].

2. Тестування юзабіліті. Юзабіліті виходять за межі тестування функціональності та поєднують тестування на функціональність, а також загальний досвід роботи студентів.

Тестування юзабіліті не слід змішувати з тестуванням прийнятності користувача . Незважаючи на те, що обидва важливі для успіху веб-програми. Кожен з них має дуже різну спрямованість і виконується на різних етапах життєвого циклу розробки програмного забезпечення.

Це можна зробити внутрішньо або отримати зовнішні тестери, які відповідають вашій потенційній базі студентів. Щоб знайти зовнішні тестери, ви можете використовувати такі сервіси, як Apple TestFlight для додатків, розроблених для магазину додатків.

Тестування юзабіліті включає наступні кроки:

1. Розробіть стратегію тестування, яка забезпечує всі функції вашої програми. до них належать навігація та вміст;
2. Набір учасників тесту – всередині чи зовні;
3. Проведіть тест під спостереженням експертів;
4. Проаналізуйте результати та вдосконаліть свою програму відповідно[88].

3. Тестування інтерфейсу. Тестування інтерфейсу гарантує, що всі взаємодії між веб-сервером та інтерфейсами сервера додатків працюють безперебійно. Це включає перевірку комунікаційних процесів, а також переконання, що повідомлення про помилки відображаються правильно. Подальше тестування полягає в тому, що перерви як користувачем, так і сервером обробляються правильно [88].

4. Тестування на сумісність. Забезпечення сумісності вашої програми з усіма браузерами та пристроями є ключовим кроком у тестуванні веб-додатків. Ось різні елементи тестування на сумісність:

– сумісність браузера. Переконайтесь, що ваша програма функціонує

правильно у різних браузерах. Сюди входить перевірка того, що JavaScript, AJAX, WebSockets, сповіщення браузера та запити автентифікації працюють як розроблено. Крім того, щоб перевірити, чи працює ваша програма у всіх браузерах (так, навіть у Internet Explorer!), Ви також повинні перевірити її на різні версії браузерів, щоб побачити, чи якісь оновлення впливають на її функціональність;

– сумісність операційної системи. Як і в різних браузерах, у ваших веб-додатків можуть виникнути проблеми в деяких операційних системах. Переконайтесь, що він працює безперебійно в Windows, macOS, Linux та Unixes;

– мобільна сумісність. У наші дні мобільність сумісність є даною. Забезпечення того, що ваша програма працює на різних пристроях та працює так само добре, як на Android, ніж на iOS, є важливою частиною веб-тестування [88].

5. Тестування продуктивності. Переконавшись у тому, що функціональність вашої програми працює належним чином та реагує на всі веб-переглядачі та пристрої, настав час ознайомитись із тим, як вона працює під великим навантаженням. Сюди входить тестування програми за різною швидкістю Інтернету та те, як вона поводиться при нормальних і пікових навантаженнях (тестування навантаження). Щоб визначити точку зламу вашої програми, вона піддається підвищенню напруги, поки вона не перестане функціонувати (стрес-тестування).

Тестування на стійкість є найважливішим видом діяльності, щоб дізнатися, як ваша програма поводиться під напругою, перш ніж ваші користувачі. Переконайтесь, що ви перевіряєте функціональність у різних сценаріях та конфігураціях обладнання та що ваша програма найкращим чином відновлюється від збоїв[88].

6. Тестування безпеки. Останній крок тестування веб-додатків гарантує, що ваша програма захищена від несанкціонованого доступу та шкідливих дій через віруси чи інше шкідливе програмне забезпечення.

Тестування безпеки веб-додатків включає такі дії:

– перевірте, чи можна отримати доступ до захищених сторінок без дозволу;

– переконайтесь, що відкрита сесія закрита після бездіяльності користувача;

– перевірте SSL програми;

– переконайтесь, що обмежені файли не можуть бути завантажені без належної авторизації.

Загалом, на цьому етапі корисний контрольний список тестування безпеки, оскільки він допомагає структурувати та організовувати тестування. Такий контрольний список повинен містити завдання в таких областях:

– безпечна передача;

– аутентифікація;

– управління сесіями;

– авторизація;

– криптографія;

– перевірка даних;

– відмова від служби;

– тести на специфічну функціональність[88].

Тестування web-додатку на базі генетичного алгоритму для задач розкрою в умовах фіксованих 2-D обмежень проводилось на комп’ютері під управлінням операційної системи Microsoft Windows 10.

Тестування проведено в браузерах:

– Google Chrome 83;

– Opera 68;

– Mozilla Firefox 76;

– Microsoft Edge.

Було проведено кроссбаузерне тестування додатку, тестування зручності використання та верстки.

При тестування зручності використання web-додатку було в першу чергу звернено увагу на навігації. Вона є досить зручною та зрозумілою. Лівий колонтитул сторінки має навігаційне меню, яке відображається на усіх сторінках додатку.

Завдяки правильно розробленій структурі додаток є простим у використанні. Контент, структура, кольорове оформлення додатку відповідає тематиці та цілям, для досягнення яких він призначений. Необхідно звернути увагу на те, що назви гіперпосилань відповідають назвам сторінок, у тексті відсутні граматичні та орфографічні помилки.

Під час виконання крос-браузерного тестування web-додатку на базі генетичного алгоритму для задач розкрою в умовах фіксованих 2-D обмежень,було перевірено властивість відображатися та працювати у браузерах GoogleChrome, Opera, MozillaFireFox, Internet Explorer ідентично. Під ідентичністю розуміється відсутність розвалів верстки та здатність відображати матеріал з однаковим ступенем читабельності. Під час тестування перевірено наступне: наявність логотипу, головного меню; відображення шрифту тексту, кольорової гами елементів; коректне відображення кнопок, блоків меню; зміна виду кнопок при наведенні; головне меню на всіх сторінках; відсутність пошкоджених посилань, граматичних помилок.

Під час виконання функціонального тестуванняweb-додаткубуло перевірено: тестування роботи всіх обов'язкових функцій додатку; перевірка валідації полів.

При тестуванні web-додаток показав свою абсолютну працездатність.Навантаження додатку проводилось на рівні з іншими аналогічними додатками.Тестування даного сервісу не виявило недоліків, було успішно виконано всі чек-листи.

З чек-листами результатів тестування можна ознайомитись у Додатку Д.

4.3 Введення в експлуатацію

Замовником даної розробки є компанія «VlaVirchenko», що спеціалізується на виробництві корпусних меблів. Підприємство ФОП Вірченко А. Г. займається виробництвом меблів з 2007 року. Пропонується індивідуальне виробництво корпусних меблів відповідно до побажань клієнтів та ескізів дизайнерів. Компанія пропонує індивідуальні дизайнерські та високоякісні індивідуальні меблі за розумними цінами. Компанія працює у різних сферах. Вони пов'язані з меблями.

Весь асортимент можна розділити на три основні напрямки:

– для офісу: столи, стільці, стелажі, полиці, вбиральні, шафи;

– для дому: кухні, шафи купе, столи обідні, стільниці, полиці для зберігання;

– виготовлення та продаж фасадів, дзеркал, вітражів та ін.

Замовлення змінюються залежно від сезону. Так влітку, більше купують саме домашні меблі на кухню або столи та стільці, а ось у сезон підвищення бізнес активності, восени, зростає попит на меблі для офісів.

Весь етап технології виготовлення можна розділити на кілька кроків:

– одержання замовлення від клієнта. Промальовування ескізу та моделювання майбутнього виробу за допомогою комп'ютерної програми;

– сировину надходить на верстат, де його розрізають на щити потрібного розміру, отриманий із програми;

– конструктивні частини виробу подаються на кромкооблицювальний верстат. Де відбувається їх зачистка та приклеювання під пресом ламінованого матеріалу. Зазвичай при цьому використовують ПВХ плівку;

– свердління отворів під кріплення та фурнітуру. Після чого відбуватиметься шліфування для зняття надлишків;

– складання готової конструкції та усунення недоліків.

На цьому виробничий етап закінчується, і меблі надходять у продаж.

Мета компанії –задоволений клієнт, компанія намагається не відставати від часу та світових трендів. Кращі фахівці працюють над створенням індивідуального дизайну для кожного клієнта, продумують кожну частину до найдрібніших деталей, створюють зручні та неповторні меблі. Наймодніші тенденції, використання сучасних технологій, ретельний відбір комплектуючих, багатоетапна система контролю якості – все це дає можливість створювати меблі, які не залишить байдужим жодного покупця. Кількість клієнтів швидко збільшується, понад 70% клієнтів стають постійними і рекомендують дану компанію друзям і знайомим.

Великий досвід фахівців дозволяє надати клієнту відмінні рішення для створення меблів, які не тільки без проблем впишеться в дизайн будь-якого приміщення, але й сумлінно слугуватимуть понад десяток років. У виробництві використовуються лише перевірені матеріали та компоненти, що гарантують бездоганну роботу механічних деталей на весь термін служби виробу.

Контроль якості виробництва виконується на кожному етапі роботи – від проектування меблів до їх складання. Так як всі меблі створюється виключно на замовлення, всі рішення –унікальні і є єдиним варіантом.

Основними критеріями розробленого додатку є:

– можливість додавання деталей з фіксованим розміром;

– можливість додавання нових листів матеріалу їх кількість та ціну за погонний метр;

– вивід діаграми оптимізованого розкрою матеріалу на основі генетичного алгоритму;

– прорахування сумарної площі листа матеріалу, деталей та відходу;

– прорахування ціни використаного матеріалу;

– автоматичне формування звіту про результати роботи над проектом.

Далі представлена демонстрація роботи розробленого додатку:

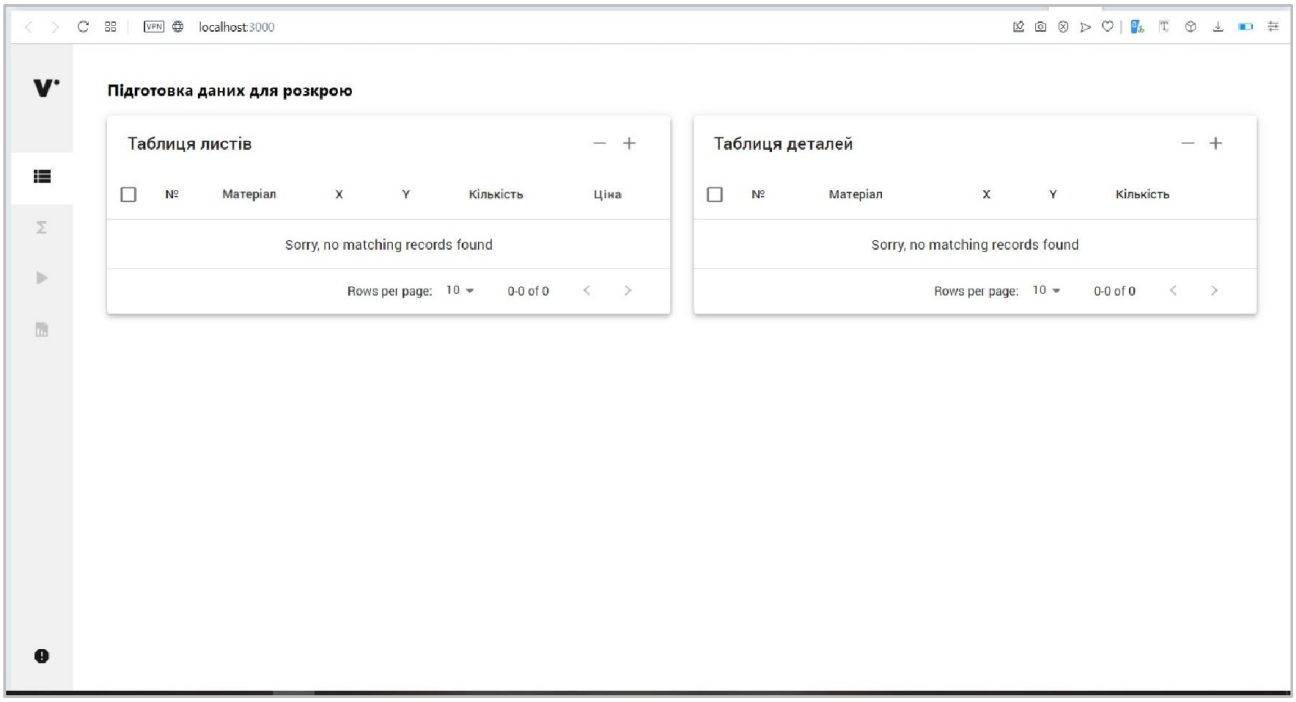


Рисунок 4.5 – Початкова сторінка програми.

Для первинної демонстрації роботи програми взято реальний проект комп’ютерного столу створений в програмі PRO100[89]. На рисунках 4.6 – 4.7 зображений проект«Комп’ютерний стіл» в програмі PRO100:

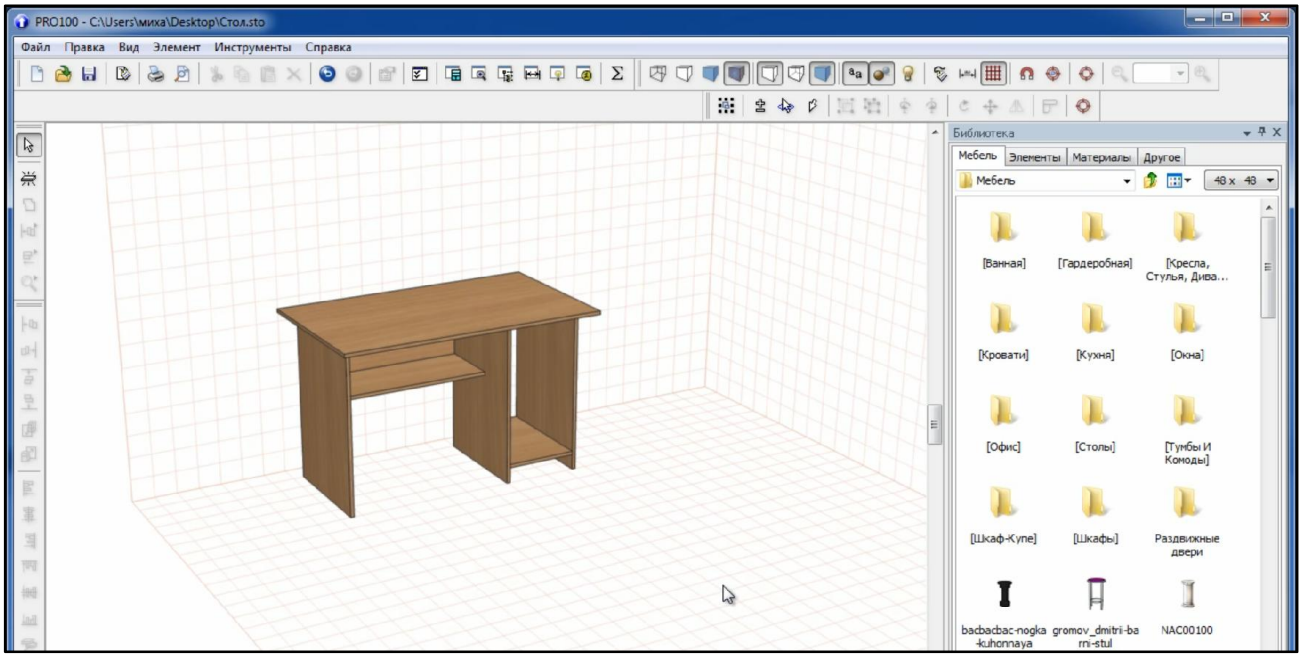


Рисунок 4.6 – Проект 1 в програмі PRO100.

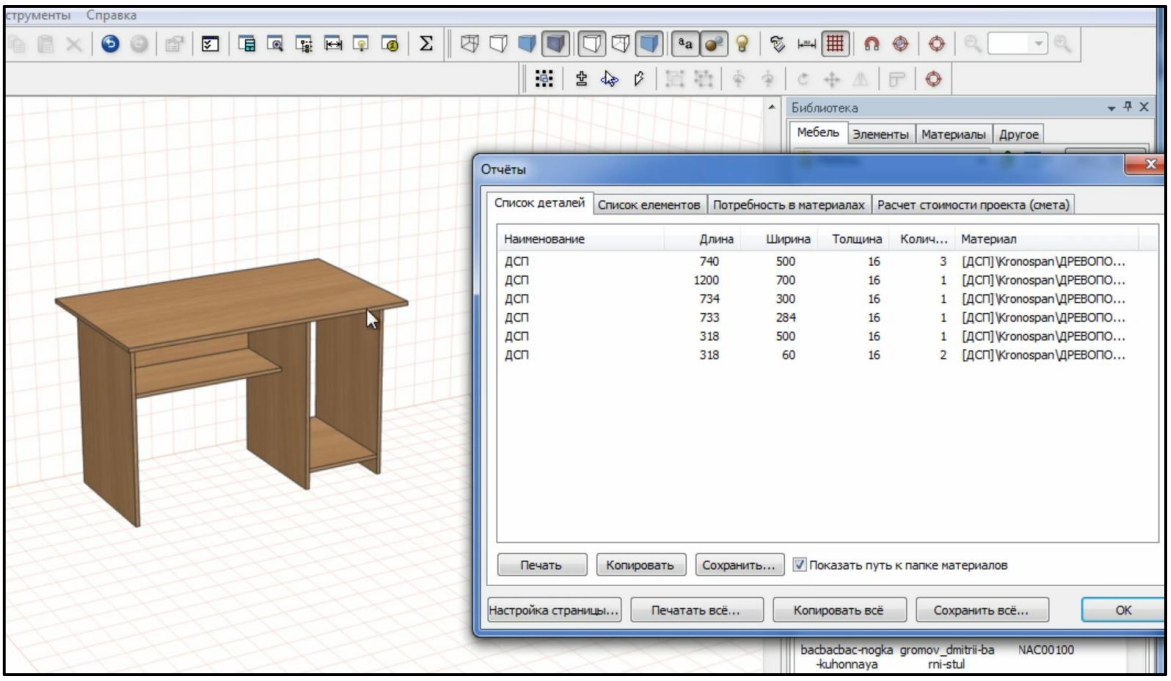


Рисунок 4.7 – Розміри деталей для проектування діаграми розкрою проекту 1.

Визначено, що проект має 9 деталей різних розмірів, для розміщення лекал достатньо одного листа матеріалу. Маємо точні розміри деталей та їх кількість.На рисунку 4.8 зображенийпроцес занесення інформації щодо листа матеріалу та деталей в додаток для прорахунку.

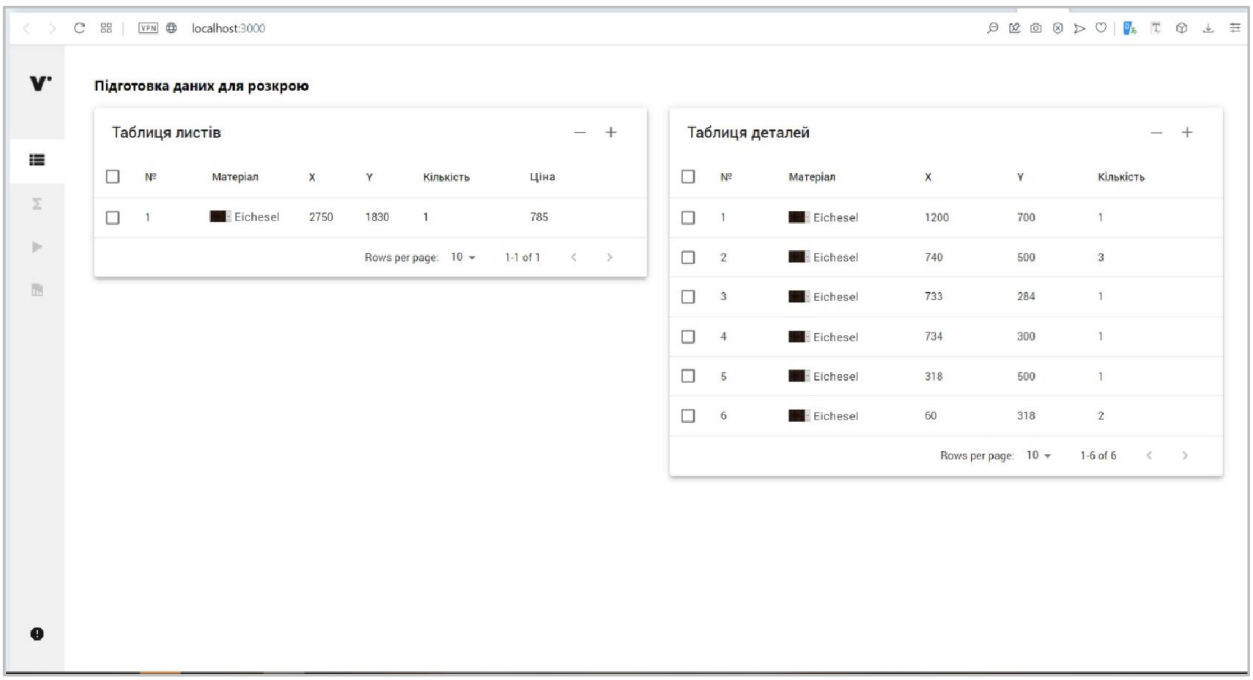


Рисунок 4.8 – Підготовка даних для розкрою проекту 1.

Рисунок 4.9 – Сумарні площі листа, деталей та відходу проекту 1.

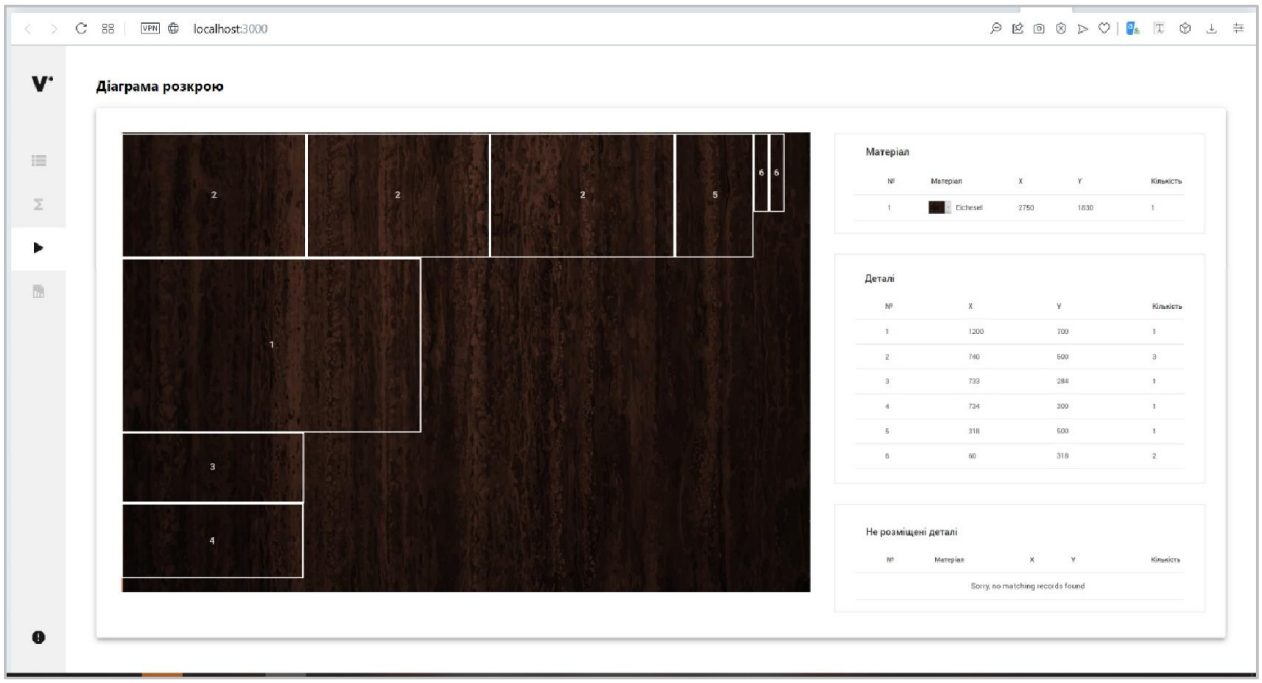


Рисунок 4.10 – Діаграма розкрою проекту 1.

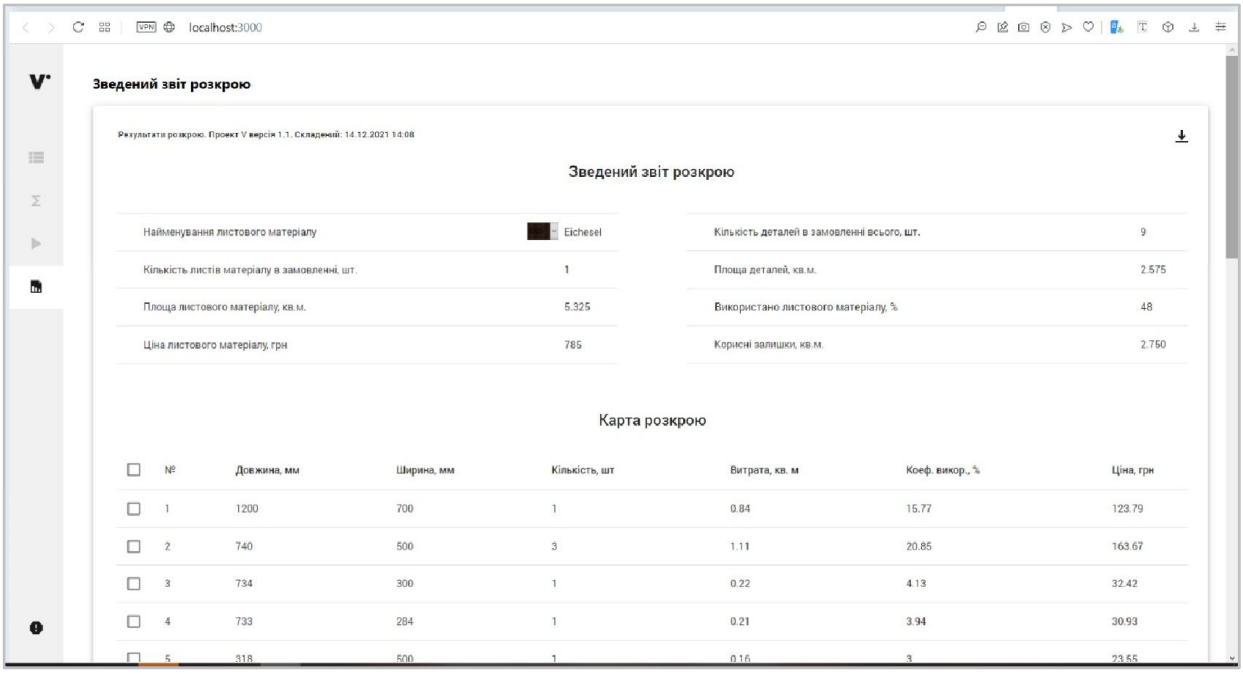


Рисунок 4.11 – Сформований звіт проекту 1.

Для більш детального розгляду роботи додатку було обрано ще один більш детальний проект. Взято реальний проект кухні створений в програмі PRO100[88]. На рисунку 4.12 зображений проект«Кухня» в програмі PRO100:

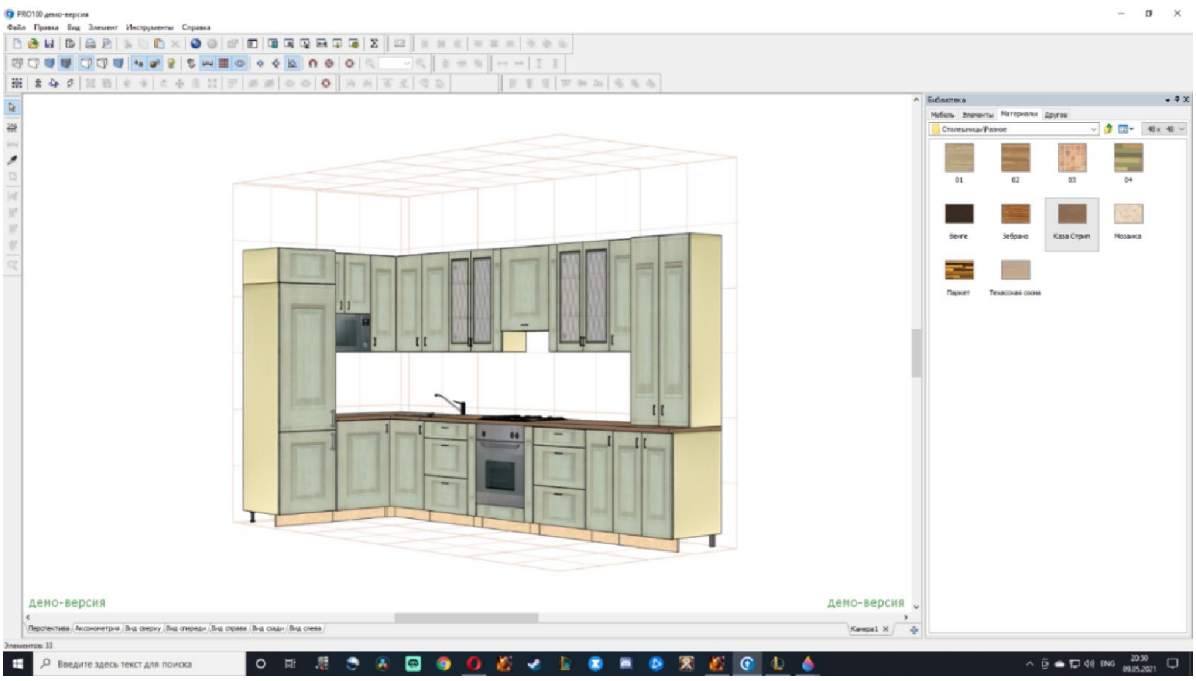


Рисунок 4.12 – Проект 1 в програмі PRO100.

Визначено, що проект має 141 деталь різних розмірів, для розміщення лекал достатньо 10 листів матеріалу. Маємо точні розміри деталей та їх кількість.На рисунку 4.13 –4.16 зображенийпроцес занесення інформації щодо листа матеріалу та деталей в додаток для прорахунку.

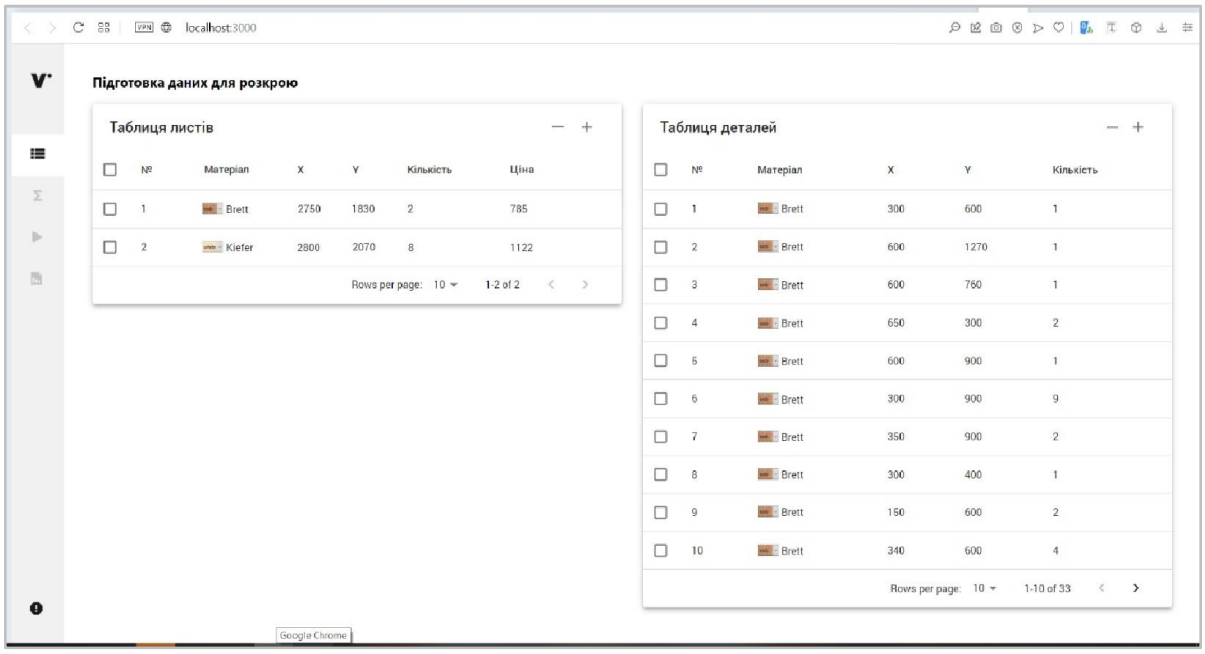


Рисунок 4.13 – Підготовка даних для розкрою проекту 2.

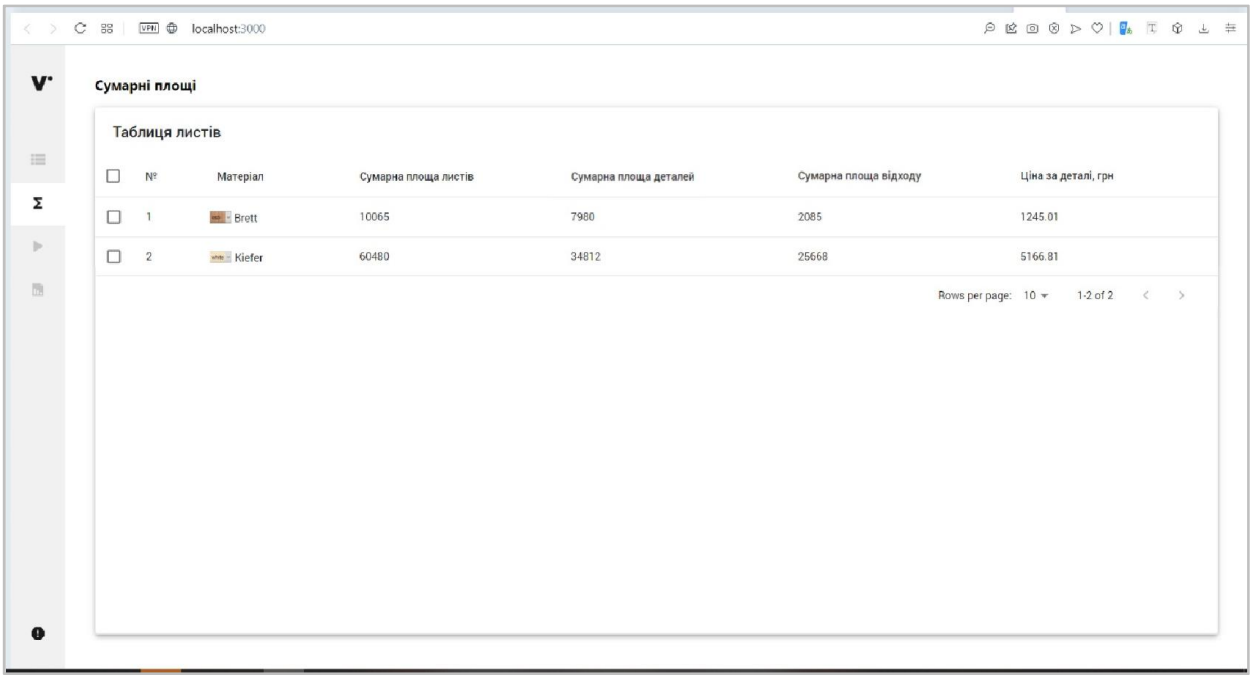


Рисунок 4.14 – Сумарні площі листа, деталей та відходу проекту 2.

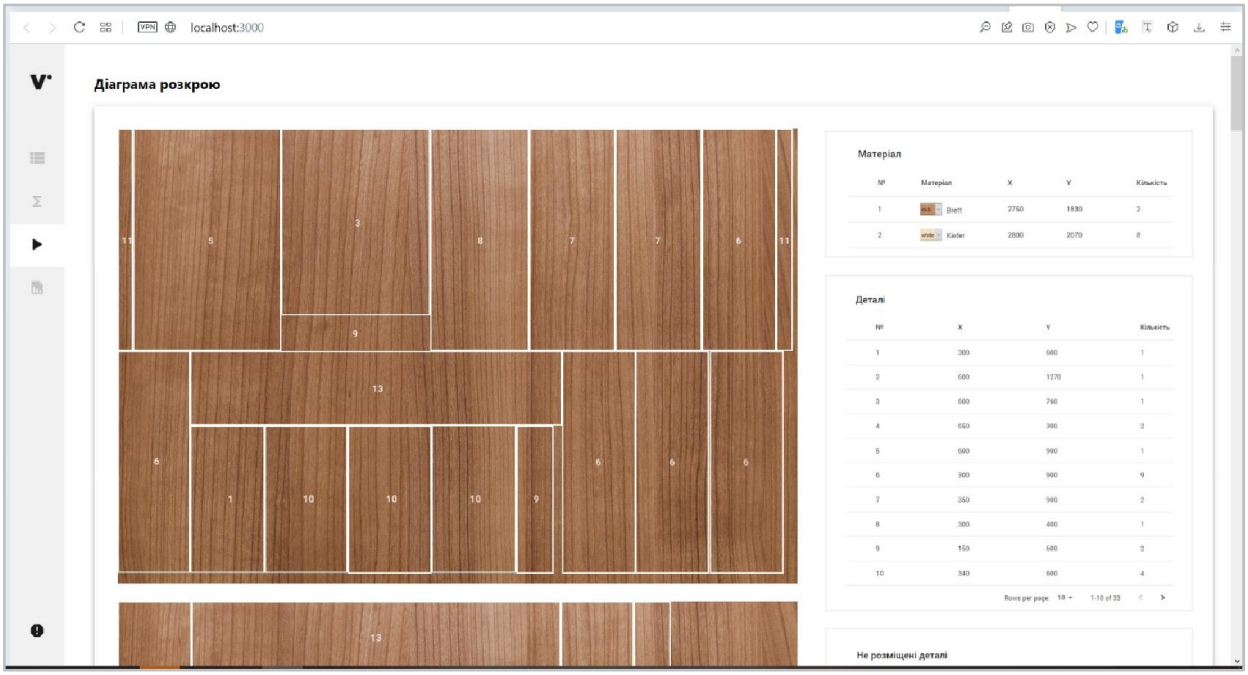


Рисунок 4.15 – Діаграма розкрою проекту 2.

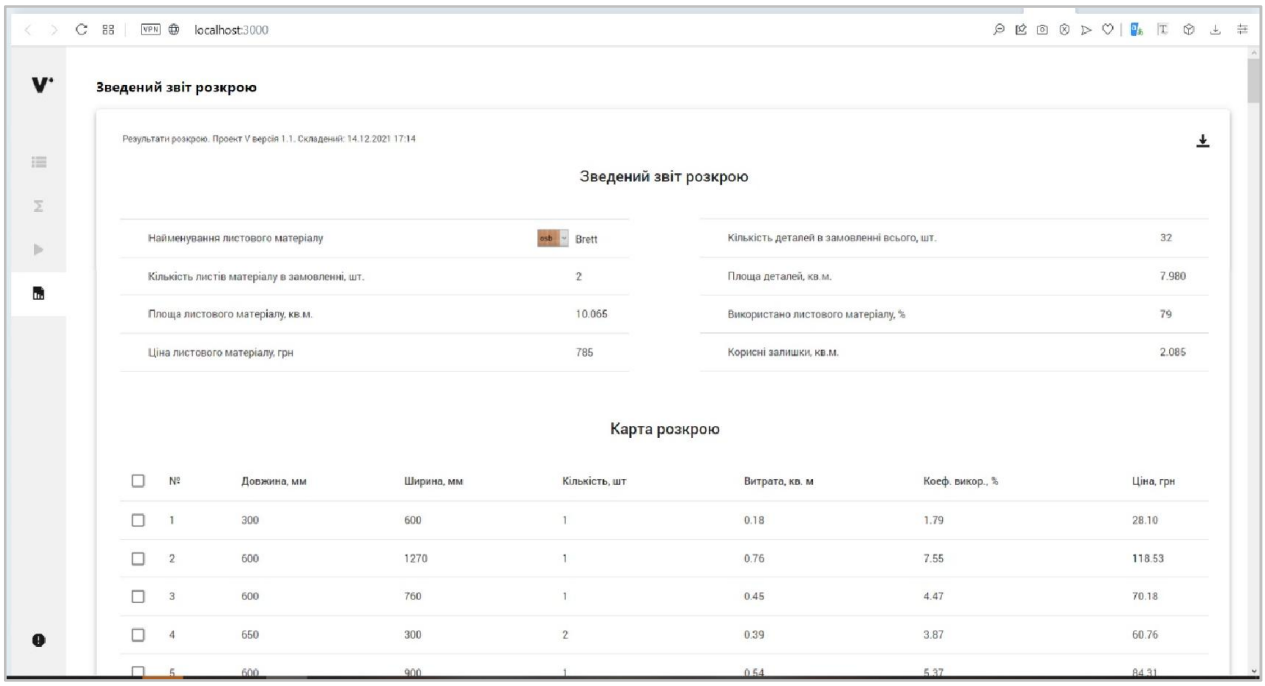


Рисунок 4.16 – Сформований звіт проекту 2.

Для розгляду результатів роботи додатку було сформовано таблицю порівняння продуктивності розроблених проектів:

Таблиця 4.1 – Порівняння продуктивності додатку

Критерії порівняння	Проект 1 «Комп'ютерний стіл»	Проект 2 «Кухня»
Кількість листів	1	10
Кількість деталей	9	141
Час прорахунку зведеної площі та ціни	2с	5с
Час формування розкрою	3с	38с
Час формування звіту	5с	12с
Не розміщено деталей	0	0
Коефіцієнт використання	48	79
Площа корисних залишків (м)	2.750	2.085

Як бачимо, продуктивність додатку на досить високому рівні. З довідкою про впровадження результатів даної праці в роботу компанії «VlaVirchenko» можна ознайомитись у Додатку Е.

ВИСНОВКИ

У дипломній роботі магістра розроблено web-додаток для роботи виробництва з виготовлення корпусних меблів на базі генетичного алгоритму в умовах фіксованих 2-D обмежень. Проведено UX case дослідження предметної області та цілей додатку. Визначено функціональні вимоги та структуру, логотип, макет, інтерфейс, спроектовано та проведено тестування власного ГА, виконано комплексне тестування та введено в експлуатацію. Було проведено ґрунтовний огляд наукових досліджень в області рішення задач 2-D розкрою. Розглянуто основні види ГА, проаналізовано предметні області для реалізації продуктів, досліджено класифікації, методи та геометричні аспекти підходів до рішення задач розкрою. При розробці були обрані та обґрунтовані веб-технології для розроблення програмного продукту, такі як HTML, JavaScript, TypeScript, Visual Studio Code що дозволяють створювати інтерактивні веб-сторінки.

Готовий додаток забезпечує наступні функції:

– можливість додавання деталей з фіксованим розміром;

– можливість додавання нових листів матеріалу їх кількість та ціну за погонний метр;

– вивід діаграми оптимізованого розкрою матеріалу на основі генетичного алгоритму;

– прорахування сумарної площі листа матеріалу, деталей та відходу;

– прорахування ціни використаного матеріалу;

– автоматичне формування звіту про результати роботи над проектом;

– можливість перегляду короткої довідки щодо роботи з додатком.

Отже розроблений додаток відповідає всім вимогам, поставленим на етапі постановки завдання, а генетичний алгоритм є ефективним.

В якості подальшого вдосконалення додатку, можна більш детально

пропрацювати функції додатку та додати нові:

- створення бази даних користувачів;
- зберігання історії проектів;
- можливість імпортувати дані з таблиці Excel чи текстового файлу;
- функція злиття замовлень;
- розробка бази стандартних лекал меблів;
- додавання налаштування товщини різу;
- можливість ручного редагування діаграми розкрою;
- створення бази даних найбільш використовуваних листів матеріалу;
- розробка панелі адміністратора для додавання нових матеріалів;
- додавання функції збереження лишків проектів для подальшого використання в наступних проектах;
- чат-підтримка з розробником додатку;
- доопрацювання інтерфейсу.

СПИСОКВИКОРИСТАНИХДЖЕРЕЛ

1. Dyckhoff H. A typology of cutting and packing problems / European Journal of Operational Research. – 1990. – № 44. – P. 150-152.

2. Rathee S, Ratnoo S, Ahuja J. (2017) “Instance selection using multi-objective CHC evolutionary algorithm.” In: Proceedings of Third International Conference on ICTCS 2017, Udaipur, India: Springer 475-484.

3. Darrel Whitley "A Genetic Algorithm Tutorial", 1993.

4. Li, G., Liu, P., Le, C., et al. (2019) A Novel Hybrid Meta-Heuristic Algorithm Based on the Cross-Entropy Method and Firefly Algorithm for Global Optimization. Entropy,21.

5. E. Skakalina. Hybridization of the genetic algorithm with the apparatus of fuzzy sets // Fourth International Scientific and Technical Conference"computer and information systems and technologies " April 22-23, 2020, Kharkov - Riga - Kiev - Lviv - P.10-11.http://csitic.nure.ua/issue/view/12193/showToc.

6. Cohoon J.P., Karro J., Lienig J. Evolutionary algorithms for the physical design of VLSI circuit. In: Advances in Evolutionary Computing:Theory and Applications. A. Ghosh, S. Tsutsui (Eds.). Springer Verlag, London, 2003, pp. 683–712.

7. Alpert Ch.J., Dinesh P., Mehta D.P., Sapatnekar S.S. Handbook of algorithms for physical design automation. CRC Press, NY, USA, 2009.

8. Ershov N.M. Non-uniform cellular genetic algorithms // Computer Research and Modeling, 2015, vol. 7, no. 3, pp. 775-780.

9. Rojas I., Gonzalez J., Pomares H., Merelo J.J., Castillo P.A., Romero G. Statistical analysis of the main parameters involved in the design of a genetic algorithm // IEEE Trans. on Systems, Man, and Cybernetics. Part C. –2002. –32, N 1. –P. 31–37.

10. Alfian Akbar Gozali and Shigeru Fujimura. Localization strategy for island model genetic algorithm to preserve population diversity. In Roger Lee, editor,

Computer and Information Science, chapter Studies in Computational Intelligence, pages 149–161. Springer International Publishing, Cham, 2018.

11. Alba, E. and Dorronsoro, B. (2008). Cellular Genetic Algorithms. Springer Publishing Company, Incorporated, 1st edition.

12. Гладков Л.А. Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М.: Физматлит, 2010. – 368 с.

13. Орлов А.Н., Курейчик В.В., Кудрякова Т.Ю. Комбинированный алгоритм решения задачи прямоугольного раскроя // Труды Конгресса по интеллектуальным системам и информационным технологиям «IS&IT'15». Научное издание в 3-х т. Т. 3. –2015. – С. 212-217.

14. SkakalinaElena. Modificationofantcoloniesalgorithmforsolvingtheproblemofautomationscheduling[Електронний ресурс] / E. Skakalina // Матеріали XV міжнародноїконференції "Контроль і управління в складних системах (КУСС-2020)", м. Вінниця, 8-10 жовтня 2020 р. – Електрон. текст. дані. – Вінниця : ВНТУ, 2020. – Режим доступу: http://ir.lib.vntu.edu.ua//handle/123456789/30669

15. Skakalina E.,development of a medium for optimizing cargo transportation using genetic algorithms // Results of modern scientific research and development. Proceedings of the 6th International scientific and practical conference. Barca Academy Publishing. Madrid, Spain. 2021. Pp. 136-143. URL: https://sci-conf.com.ua/vi-mezhdunarodnaya-nauchno-prakticheskaya-konferentsiya-results-of-modern-scientific-research-and-development-22-24-avgusta-2021-goda-madrid-ispaniya-archive /.

16. О.В.Скакаліна, С.С.Носенко. Двох-етапнаоптимізаціяіззастосуваннямзасобівуправлінняпроектамитагенетичнихалгоритмів. Наукова конференція професорів, викладачів, наукових працівників, аспірантів та студентів університету 15 квітня – 15 травня 2016 року. - Полтава: ПолтНТУ, 2016.- Том 2, с.95-96.

17. Skakalina Elena. Generation of optimal 2-d routes using genetic algorithms // scientific trends and trends in the context of globalization. Proceedings

of the 2nd International Scientific and Practical Conference "Scientific Trends and Trends in the Context of Globalization" (September 19-20, 2021). Umeå, Sweden: Mondial, 2021. https://interconf.top/archive-9.html.

18. Скакаліна О. В. , Клочко А. О. ЗАСТОСУВАННЯ GENETIC ALGORITHM TOOL ПРИ ОПТИМІЗАЦІЇ ПЕРЕВЕЗЕНЬ ПРОДУКЦІЇ В АХ «АСТАРТА» / Олена Скакаліна // Матеріали V Всеукраїнської науково-технічної конференції «Комп'ютерна математика в науці, інженерії та освіті CMSEE-2020» (27 листопада 2020 року, м. Полтава) .- Полтава: Національний університет «Полтавська політехніка імені Юрія Кондратюка», 2020.- С. 70-72.

19. Скакаліна О. В., Клочко А. О. СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОЇ WEB-ПЛАТФОРМИ ДЛЯ ЦИФРОВІЗАЦІЇ ЛОГІСТИКИ / Олена Скакаліна // ЗБІРНИК НАУКОВИХ ПРАЦЬ за матеріалами XIII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2021» 15-16 жовтня 2021 р. - Хмельницький: - 2021. С. – 214-219.

20. Kuliev E.V. Dukkardt A.N, Kureychik V.V. Legebokov A.A. Neighborhood Research Approach in Swarm Intelligence for Solving the Optimization Problems // Proceedings of IEEE EastWest Design & Test Symposium – (EWDTS'2014) Kiev, Ukraine, September 26–29, 2014. – P. 112-115.

21. Раціональний розкрій промислових матеріалів / Л.В. Канторович,В.А. Залгаллером. - Новосибірськ: Наука СО, 1971. - 320 с.

22. Dyckhoff, H. A typology of cutting and packing problems / H. Dyckhoff // European Journal of Operation Research. – 1990. – Vol. 44. – P. 145-159.

23. Dyckhoff, H. Cutting and packing. / H. Dyckhoff, G. Scheithauer, J. Terno // In: Annotated Bibliographies in Combinatorial Optimization, M. Dell'Amico, F. Maffioli, S. Martello (eds.). – John Willey & Sons. – 1997. – P. 393-412.

24. Wäscher, G. An improved typology of cutting and packing problems / G. Wäscher, H. Haußner, H. Schumann // European Journal of Operational Research. – 2007. – Vol. 183, № 3. – P. 1109-1130.

25. Morabito, R. A simple and effective recursive procedure for the manufacturer's pallet loading problem / R. Morabito, S. Morales // Journal of the Operational research Society. – 1998. – Vol. 49. – P. 819-828.

26. Євсєєва, Л.Г. Завданняупаковкиінтервальнихкіл / Л.Г. Євсєєва, Г.Н. Яськов //Східно-Європейськийжурналпередовихтехнологій. - 2008. - Т. 1, № 4. - С. 25-29.

27. Ratcliff, M.S.W. Allowing for weight considerations in container loading / M.S.W. Ratcliff, E.E. Bischoff // OR Spektrum. – 1998. – Vol. 20. – P. 65-71.

28. Kellerer, H. Multidimensional knapsack problem / H. Kellerer, U. Pferschy, D. Pisinger // Knapsack problems. – Springer, Berlin, Heidelberg, 2004. – P. 235-283.

29. Martello, S. Knapsack Problems – Algorithms and Computer Implementations / S. Martello, P. Toth. – Chichester: John Wiley & Sons, 1990.

30. Erlebach, T. Approximating multiobjective knapsack problems / T. Erlebach, H. Kellerer, U. Pferschy // Management Science. – 2002. –Vol. 48, № 12. – P. 1603-1612.

31. Skakalina E.V. IMPLEMENTATION OF A GENETIC ALGORITHM FOR OPTIMIZING THE DISTRIBUTION PROCESS / Elena Skakalina // Modern engineering and innovative technologies. – 2020. – Issue 14, Part. 2, pp. 6-13. ISSN 2567-5273, DOI: 10.30890/2567-5273.2020-14-02.

32. Martello, S. An exact approach to the strip-packing problem / S. Martello, M. Monaci, D. Vigo // INFORMS Journal on Computing. – 2003. – Vol. 15, № 3. – P. 310-319.

33. Scheithauer, G. A three-dimensional bin packing algorithm / G. Scheithauer // Elektronische Informationsverarbeitung und Kybernetik. – 1991. – Vol. 27, № 5/6. – P. 263-271.

34. Milenkovic, V.J. Translational polygon containment and minimal enclosure using mathematical programming / V.J. Milenkovic, K. Daniels //

International Transactions in Operational Research. – 1999. – Vol. 6, № 5. – P. 525-554.

35. Wäscher, G. Heuristics for the integer one-dimensional cutting stock problem: A computational study / G. Wäscher, T. Gau // Operations-Research-Spektrum. – 1996. – Vol. 18, № 3. – P. 131-144.

36. Gilmore, P.C. Multistage cutting stock problems of two and more dimensions / P.C. Gilmore, R.E. Gomory // Operations Research. – 1965. – Vol. 13, № 1. – P. 94-120.

37. Марков, В.Н. База знань для негільотінного розкрою-упаковки / В.Н. Марков //Інформаційні технології. - 2007. - № 4. - С. 17-23.

38. Козин, І.В. Генерація випадкових карт прямокутного розкрою / І.В. Козин,С.Є. Батовського // Питання прикладної математики и математичного моделювання. -2018. - № 18. - С. 118-126.

39. Martello, S. The three-dimensional bin packing problem / S. Martello, D. Pisinger, D. Vigo // Operations Research. – 2000. – Vol. 48, № 2. – P. 256-267.

40. Ульянов, М.В. Комбінованийіхвильовоїалгоритмирозв'язаннязадачіупаковки: принципипобудовитаособливості / М.В. Ульянов, О.А. Наумова // Бізнесінформатіка. - 2009. - № 2. - С. 27-33

41. Lodi, A. Models and bounds for two-dimensional level packing problems / A. Lodi, S. Martello, D. Vigo // Journal of Combinatorial Optimization. – 2004. – Vol. 8, № 3. – P. 363-379.

42. Martello, S. Exact solution of the two-dimensional finite bin packing problem / S. Martello, D. Vigo // Management Science. – 1998. – Vol. 44. – P. 388-399.

43. Теоріярозкладів. Завданняіалгоритми / А.А. Лазарєв, О.Р. Гафаров. - М .: МГУ, 2011. - 222 с.

44. Coffman, E.G. Approximation algorithms for bin-packing–an updated survey / E.G. Coffman, M.R. Garey, D.S. Johnson // Algorithm design for computer system design. – Springer, Vienna, 1984. – P. 49-106.

45. Burke, E.K. A new placement heuristic for the orthogonal stock-cutting problem / E.K. Burke, G. Kendall, G. Whitwell // Operations Research. – 2004. – Vol. 52, № 4. – P. 655-671.

46. Мухачева, Е.А. Методи локального пошуку в задачах ортогонального розкрою іупаковки: аналітичний огляд і перспективи розвитку / Е.А. Мухачева [и др.] //Інформаційні технології. - 2004. - № 5. Додаток. - С. 2-17.

47. Генетичні алгоритми / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик; під ред.В.М. Курейчика. - 2-е изд., Испр. і доп. - М .: ФИЗМАТЛИТ, 2006. - 320 с.

48. Корчевська, О.В. Рішення задач ортогонального розкрою-упаковки на основіконструктивних і нейромережевих підходів / О.В. Корчевська // Освітніресурси і технології. - 2014. - № 1 (4). - С. 142-149.

49. Gonzalez, T.F. Handbook of approximation algorithms and metaheuristics / T.F. Gonzalez. – Chapman and Hall/CRC, 2007. – 1432 p.

50. E. Skakalina, A. Klochko . PRINCIPLES OF FORMATION OF GENERATIONS IN GENETIC ALGORITHMS / Elena Skakalina // LXXIII – International Scientific Conference "EUROPEAN INTEGRATION IN SCIENCE AND INNOVATIONS", Chernivtsi, December 15-16, 2020.- P. 20-22.

51. Ягудін, Р.Р. Оптимізація компонування тривимірних геометричних об'єктів на основі годографа вектор-функції щільного розміщення / Р.Р. Ягудін // Інженерний вісник Дона. - 2012. - Т. 21, № 3. - С. 206-217.

52. Bennell, J. Tools of mathematical modeling of arbitrary object packing problems / J. Bennell [et al.] // Annals of Operations Research. – 2010. – Vol. 179, № 1. – P. 343-368.

53. Stoyan, Y. Optimized object packings using quasi-phi-functions / Y. Stoyan [et al.] // Optimized packings with applications. – Springer, Cham, 2015. – P. 265-293.

54. Стоян, Ю.Г. Моделюваннящільноїупаковки 3D-об'єктів / Ю.Г. Стоян, В.В. Сьомкін, А.М. Чугай // Кібернетикаісистемнийаналіз. - 2016. - С. 137-146.

55. Програма «Об'ємник – меблеве підприємство» [Електронний ресурс]. – Режим доступу: https://mebelsoft.org

56. Програма «Астра Розкрій» [Електронний ресурс]. – Режим доступу: http://astrapro.ru/default.asp?page=astra-raskroj

57. Програма «Catting 3» [Електронний ресурс]. – Режим доступу: https://www.softportal.com/software-26387-cutting-3.html

58. Програма «Woody» [Електронний ресурс]. – Режим доступу: http://www.woodypro.ru

59. Програма «Sawyer» [Електронний ресурс]. – Режим доступу: https://gidmaster.info/soft.php?text_20120808000000

60. Управління проектами [Електронний ресурс]. – Режим доступу: https://www.businessstudio.ru/articles/article/

61. Ітеративна модель розробки [Електронний ресурс]. – Режим доступу: https://web-creator.ru/articles/iterative_development

62. Customer Journey Map [Електронний ресурс]. – Режим доступу: https://www.uplab.ru/blog/customer-journey-map/

63. Програма «Mindomo» [Електронний ресурс]. – Режим доступу: https://www.mindomo.com

64. Програма «Miro» [Електронний ресурс]. – Режим доступу: https://miro.com/app/dashboard/

65. Як розрахувати час проекту [Електронний ресурс]. – Режим доступу: https://spravochnick.ru/freelance/kak_rasschitat_vremja_neobhodimoe_dlja_raboty_nad_proektom/

66. Програма «Jira» [Електронний ресурс]. – Режим доступу: https://www.atlassian.com/ru/software/jira

67. UML – діаграма варіантів використанні [Електронний ресурс]. – Режим доступу: https://habr.com/post/47940/

68. Програма «Draw.io» [Електронний ресурс].–https://www.draw.io

69. Діаграма станів [Електронний ресурс]. – Режим доступу: https://studfiles.net/preview/5010027/page:5/

70. Трутнев Д. Р., Архітектури інформаційних систем. Основи проектування.: учбов. посіб / Трутнев Д. Р. – Санкт-Петербург: НИУ ИТМО, 2012. – 66 с.

71. Пономаренко В. С., Проектування інформаційних систем: посібник. / Пономаренко В. С., Пушкар О. І., Журавльова І. В., Мінухія С. В. – Київ: Видавничий центр «Академія», 2002. – 488 с.

72. Кольори у web-дизайні: Вибір правильного поєднання для вашого сайту [Електронний ресурс]. – Режим доступу:https://habr.com/post/105250/

73. Психологія кольору у дизайні сайтів [Електронний ресурс]. – Режим доступу:https://depix.ru/articles/psihologiya_tsveta_i_web_dizayn

74. Шрифт «Roboto»[Електронний ресурс]. –Режим доступу:https://fonts.google.com/specimen/Roboto?subset=cyrillic#standard-styles

75. Створення логотипу [Електронний ресурс]. – Режим доступу: http://artdeco.in.ua/nashi-uslugi/poligrafiya-pechat/sozdanie-logotipa/

76. Дизайн навігація: вертикальне меню [Електронний ресурс]. – Режим доступу: https://www.sokolovm.com/blog/dizayn-navigacii-vertikalnoe-menyu

77. Траєкторія переглядів F-Shaped Pattern [Електронний ресурс]. – Режим доступу: https://netpeak.net/ru/blog/kak-uluchshit-ux-sayta-12-sovetov-po-yuzabiliti/

78. Програма «Figma»[Електронний ресурс]. – Режим доступу: https://www.figma.com

79. Програма «Photoshop»[Електронний ресурс]. – Режим доступу: https://www.adobe.com/ua/products/photoshop.html

80. Переваги веб-сервісів [Електронний ресурс]. – Режим доступу: https://bigbird.ru/faq/v-chem-zaklyuchayutsya-preimushchestva-veb-servisa-i-v-chem-ego-otlichie-ot-nastolnyh-programm

81. Стівен Шафер. HTML, XHTML і CSS. Біблія користувача, 5-е видання = HTML, XHTML, and CSS Bible, 5th Edition. - М .: «Діалектика», 2010. - 656 с.

82. Alexei White. Major JavaScript Engines // JavaScript Programmer's Reference. –Indianapolis, IN 46256: Wiley Publishing, Inc., 2009. –P. 12-13. – (Programmer's Reference). –ISBN 978-0-470-34472-9.

83. TypeScript [Електронний ресурс]. – Режим доступу: https://www.typescriptlang.org.

84. Програма «VisualStudioCode»[Електронний ресурс]. – Режим доступу: https://code.visualstudio.com.

85. Скакаліна О. В. , Клочко А. О. Застосування генетичного алгоритму для вирішення задачі розкрою комплектуючих для корпусних меблів// Тези 73-ої наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів Національного університету «Полтавська політехніка імені Юрія Кондратюка» 21 квітня – 13 травня 2021 р. - Національний університет «Полтавська політехніка імені Юрія Кондратюка», Полтава: - 2021.- Том 1, с. 456-458.

86. GladkovL.A., KurejchikV.V., KurejchikV.M. Geneticalgorithms, 2010.

87. Global Optimization Test [Електронний ресурс]. – Режим доступу: https://www.mat.univie.ac.at/~neum/glopt/test_results.html.

88. Навчальний посібник з дисципліни «Методи тестування і оцінки якості програмного забезпечення» для студентів денної та заочної форми навчання: 6.050101 – «Комп’ютерні науки та інформаційні технології». Укл.: Колектив провідної української компанії з тестування програмного забезпечення QATestLab, О.Л. Ляхов, О.О. Бородіна. / Полтава: ПолтНТУ, 2015 – 372 с.

89. Програма«PRO100» [Електронний ресурс]. – Режим доступу: http://spb-pro100.ru/vazhno/o_programme_pro100/.

ДОДАТОК А
ІНФОРМАЦІЯ ПРО НАУКОВІ ПРАЦІ

№ п/п	Найменування роботи	Форма роботи	Вихідні дані	Об'єм	Автори
1	Застосування geneticalgorithmtool при оптимізації перевезень продукції в АХ «АСТАРТА»	Друкована	Матеріали V Всеукраїнської науково-технічної конференції «Комп'ютерна математика в науці, інженерії та освіті CMSEE-2020» (27 листопада 2020 року, м. Полтава) .- Полтава: Національний університет «Полтавська політехніка імені Юрія Кондратюка», 2020.	2 с.	Скакаліна О. В. Клочко А. О.
2	Principles of formation of generations in genetic algorithms	Друкована	LXXIII – International Scientific Conference "European integration in science and innovations", Chernivtsi, December 15-16, 2020.	2 с.	E. Skakalina, A. Klochko
3	Застосування генетичного алгоритму для вирішення задачі розкрою комплектуючих для корпусних меблів	Друкована	Тези 73-ої наукової конференції професорів, викладачів, наукових працівників, аспірантів та студентів Національного університету «Полтавська політехніка імені Юрія Кондратюка» 21 квітня – 13 травня 2021 р. - Національний університет «Полтавська політехніка імені Юрія Кондратюка», Полтава: - 2021.- Том 1	2 с.	Скакаліна О. В. Клочко А. О.
4	Створення інтелектуальної web-платформи для цифровізації логістики	Друкована	Збірник наукових праць за матеріалами XIII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2021» 15-16 жовтня 2021 р. - Хмельницький: - 2021.	5 с.	Скакаліна О. В. Клочко А. О.

ДОДАТОК Б

СЕРТИФІКАТ BEF

B|E|F Bukovinian Economic Foundation

CERTIFICATE

This certificate confirms that

Klochko Anastasia

fifth year student

National University «Yuri Kondratyuk Poltava Politechnics »

Poltava, Ukraine

participated actively in the work of

LXXIII International Scientific Conference

EUROPEAN INTEGRATION

IN SCIENCE AND INNOVATIONS

Chernivtsi, December 15-16, 2020

in the area of the conference

Information Technology

with the report

PRINCIPLES OF FORMATION
OF GENERATIONS IN GENETIC ALGORITHMS

Chairman of
The Bukovinian
Economic
Foundation

Lysiuk I.

ДОДАТОК В

OUTLINE ПРОЕКТУ

VlaVirchenko_2D_cut

- Загальні відомості
  - Вид проекту
    - web-додаток
  - Мова
    - UA
  - Клієнти
    - Фокус-група: люди 25-55 років
    - Робітники меблевого сегменту
    - В більшій степені споживачі з високим рівнем прибутку
    - Люди, яким потрібен швидкий 2-D розкрій
  - Концепція
    - Унікальний веб-додаток для розкрою листового матеріалу на базі оптимізованого ГА, з можливістю автоматизованого прорахунку вартості проекту, візуалізації діаграми розкрою та якісним звітом.
  - Цілі
    - розробника
      - Провести дослідження ринку та додатку
      - Розробити прототип та UI-кіт
      - Спроектувати ГА
      - Розробити додаток
      - Протестувати додаток
      - Реліз продукту
    - клієнта
      - Отримати власний унікальний продукт
      - Автоматичний прорахунок ціни використаного матеріалу
      - Якісна діаграма розкрою
      - Прорахунок відходу та використання матеріалу
      - Цідісний сформований звіт, який можна відіслати замовнику
- Генетичний алгоритм
  - Опис
    - Розкрій в умовах фіксованих 2-D обмежень
    - Критерії вибору лекал
      - максимальна площа
      - мінімальна площа
      - мінімальний периметр
      - максимальний периметр
      - мінімальна описана площа прямокутника
      - максимальна описана площа прямокутника
  - Послідовність
    - 1. Початок роботи алгоритму
    - 2. Отримання вхідних даних.
    - 3. Розраховуємо, які лекала потрапляють до певних контурів.
    - 4. Для кожного генома проводимо ітерацію
    - 5. Розраховуємо площу відходів і вибираємо m% найкращих результатів.
    - 6. Розраховуємо, які лекала потрапляють до нових контурів.
    - 7. Чи є для утворених нових контурів лекала?
    - 8. Утворені контури відносимо до відходів.
    - 9. Чи є ще лекала для розміщення?
    - 10. Схрещуємо найкращі результати і отримуємо нові можливі геноми.
    - 12. Кінець роботи алгоритму.

- Тестування
  - Тестові задачі
    - De Jong 2
    - De Jong 3
    - De Jong 5
    - Griewank
  - Відношення знайденого максимального значення до реального максимуму
  - Пристосованість функцій
  - Зміни пристосованості функцій
  - Використання різних операторів кросовера для функцій
- Підготовка даних
  - Дії з таблицями
    - Очистка таблиці листів
    - Очистка таблиці деталей
    - Додавання листа
    - Додавання деталі
    - Можливість виділення окремого листа
    - Можливість виділення окремої деталі
    - Видалення листа
    - Видалення деталі
  - Деталі
    - Номер
    - Лист
    - Довжина
    - Ширина
    - Кількість
    - Замітка
  - Листи
    - Номер
    - Назва листа
    - Довжина
    - Ширина
    - Кількість
    - Замітка
    - Ціна
- Сумарні площі
  - Номер
  - Лист
  - Сумарна площа листів
  - Сумарна площа деталей
  - Сумарна площа відходу
  - Ціна за деталі
- Діаграма розкрою
  - Діаграма розкрою
    - Візуалізований лист
    - Візуальзована деталь
    - Номер деталі
  - Інф лист матеріалу
    - Назва
    - Номер
    - Розмір
    - Кількість

- Інф деталі
  - Номер
  - Розмір
  - Кількість
- Не розміщені деталі
  - Номер
  - Розмір
  - Кількість

- Зведений звіт
  - Назва файлу
    - Назва проекту
    - Версія додатку
    - Дата
    - Час
  - Зведений звіт розкрою
    - Лист
      - Найменування листового матеріалу
      - Кількість листів матеріалу в замовленні, шт.
      - Площа листового матеріалу, кв.м.
      - Ціна листового матеріалу, грн
    - Деталь
      - Кількість деталей в замовленні всього, шт.
      - Площа деталей, кв.м.
      - Використано листового матеріалу, кв.м
      - Корисні залишки, кв.м.
  - Карта розкрою
    - Номер
    - Довжина, мм
    - Ширина, мм
    - Кількість, шт
    - Витрата, кв. м
    - Коеф. викор., %
    - Ціна, грн
    - Лист
      - Назва
      - Розмір
      - Кількість
  - Діаграма розкрою
  - Експорт

- Довідка
  - Про програму (коротко)
    - Лого
    - Версія додатку
    - Дата останнього оновлення
    - Автор додатку
    - Країна та місто створення додатку
    - Контакти автора додатку
  - Про програму (детально)
    - Призначення
    - Функції
    - Браузери

- Підготовка даних
  - Опис функціоналу
  - Опис вікна
  - План дій
- Сумарні площі
  - Опис кожного компоненту таблиці
  - Опис вікна
- Діаграма розкрою
  - Розбір алгоритму дії
  - Опис складових легенди
- Звіт
  - Можливість експорту
  - Опис компонентів звіту
    - Дата й час/версія додатку/назва проекту
    - зведений звіт розкрою
    - карта розкрою
    - Діаграма розкрою

ДОДАТОК Д
ЧЕК-ЛИСТИ ТЕСТУВАННЯ WEB-ДОДАТКУ

Таблиця А.1 – Тестування верстки

План тестування	Браузер	Браузер	Браузер
Наявність елементів сторінок			
Перевірити коректне відображення сторінок згідно файлів	Skipped	Skipped	Skipped
Перевірити наявність логотипу	Passed	Passed	Passed
Перевірити наявність головного меню	Passed	Passed	Passed
Перевірити наявність підвалу сайту	Passed	Passed	Passed
Відображення елементів			
Перевірити коректне відображення шрифтів тексту	Passed	Passed	Passed
Перевірити коректне відображення кнопок, блоків меню і т.д.	Passed	Passed	Passed
Перевірити відображення кольорової гами усіх елементів	Passed	Passed	Passed
Перевірити коректне розміщення банерів	Passed	Passed	Passed
Активні елементи	not run	not run	not run
Перевірити наявність підказок для активних елементів	Passed	Passed	Passed
Перевірити реагування активних елементів на наведення	Passed	Passed	Passed
Зміст сторінок			
Перевірити орфографію/граматику контенту сайту	Passed	Passed	Passed

Таблиця А.2 – Тестування на зручність використання

План тестування	Критерії проходження/Підпункти	Microsoft Edge	Mozilla Firefox	Google Chrome	Opera
1. Перевірка обов'язкових атрибутів сайту					
1.1. Логотип компанії		Passed	Passed	Passed	Passed
1.2. Головна сторінка (огляд за 5 секунд)	Перегляд головної сторінки, з якого стає ясно призначення сайту і основна його структура, не займає більше ніж 5 секунд	Passed	Passed	Passed	Passed
1.3. інформація про компанію, проект,		Passed	Passed	Passed	Passed
1.4. Наявність контактної інформації		Passed	Passed	Passed	Passed
2. Навігація на сайті					
2.1. Зрозумілість навігації по сайту		Passed	Passed	Passed	Passed
2.2. Зрозумілість поточного місця		Passed	Passed	Passed	Passed
2.3. Короткі назви меню		Passed	Passed	Passed	Passed
3. Оформлення елементів сторінки					
3.1. Хедер:					
	· розмір ширина/висота;	Passed	Passed	Passed	Passed
	· позиціонування елементів хедера на всіх сторінках;	Passed	Passed	Passed	Passed
	наявність/відсутність графічних елементів, їх коректні розмір і	Passed	Passed	Passed	Passed
3.2. Головне меню:					
	· наявність посилання «Головна» на всіх сторінках	Passed	Passed	Passed	Passed
	· розмір/колір/тип шрифта	Passed	Passed	Passed	Passed
	· вирівнювання тексту	Passed	Passed	Passed	Passed
	· постійні ширина/висота	Passed	Passed	Passed	Passed
	· зміна вигляду кнопок при наведенні курсора, натисканні, активації	Passed	Passed	Passed	Passed

ДОДАТОК Е
ДОВІДКА ПРО ВВЕДЕННЯ В ЕКСПЛУАТАЦІЮ

ФОП Вірченко А. Г.
37002, Полтавська обл., Лубенський р-н, м. Пирятин,
вул. Велика Васильківська, буд. 163а; ЄДР 2638406310;
тел. +38 (0509) 34-17-22, e-mail: mf_vlavi@mail.ru;
https://mfvlavi.com.ua

_____№_____ На №_____від_____

ДОВІДКА

про впровадження результатів дипломної роботи
на тему:
«Розробка web-додатку на базі генетичного алгоритму для задач розкрою
в умовах фіксованих 2-D обмежень»
студентки кафедри комп'ютерних та інформаційних технологій і систем
Національного університету «Полтавська політехніка
імені Юрія Кондратюка»

Цим підтверджується, що результати дипломної роботи на тему «Розробка web-додатку на базі генетичного алгоритму для задач розкрою в умовах фіксованих 2-D обмежень» впроваджені та використовуються в повсякденному використанні працівниками виробництва. Продукт забезпечує оптимізацію при рішенні задач 2-D розкрою і автоматизації складських операцій. Завдяки вдалій оптимізації, швидкість формування замовлень зрачно зросла.

Генеральний директор Вірченко Анатолій Григорович Ідент.номер 2638406310 **Вірченко А. Г.**