

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КІЇВСЬКИЙ ПОЛТЕХНІЧНИЙ ІНСТИТУТ»**

Факультет прикладної математики

Кафедра прикладної математики

До захисту допущено

Завідувач кафедри ПМА

_____ О. Р. Чертов

«____» _____ 2015р.

ДИПЛОМНА РОБОТА

на здобуття ступеня бакалавра

зі спеціальності 6.040301 «Прикладна математика»

на тему: Інформаційно-довідкова система підбору конфігурації комп'ютера

Виконав: студент 4 курсу, групи КМ-12

Вовченко Олексій Олегович

Керівник доцент Олефір О.С. $\sum \frac{1}{n^2}$

Консультант з нормоконтролю ст. викладач Мальчиков В.В.

Рецензент доцент Тесленко О.К.

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Вовченко О.О. _____

Київ – 2015 року

**Національний технічний університет України
«Київський політехнічний інститут»**

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти – перший(бакалаврський)

Спеціальність 6.040301 «Прикладна математика»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ О. Р. Чертов
«____» _____ 2015р.

ЗАВДАННЯ

на дипломну роботу студенту

Вовченку Олексію Олеговичу

1. Тема роботи «Інформаційно-довідкова система підбору конфігурації комп’ютера»

керівник роботи доцент Олефір О.С.

затверджені наказом по університету від "28" травня 2015 р. № 955-С.

2. Строк подання студентом роботи "15" червня 2015р.

3. Вхідні дані до роботи:

- відомості про бажані параметри;
- тип комп’ютера;
- максимальна ціна комп’ютера.

4. Зміст розрахунково-пояснювальної записки (перелік завдань, які потрібно розробити):

- Вивчити літературні джерела за тематикою транспортної логістики;
- Провести аналіз існуючих програмних рішень у даній галузі;
- Проаналізувати математичне забезпечення, що використовується при рішенні даних задач;

- Розробити програмний продукт, що буде проводити підбір оптимальної конфігурації комп’ютера;
- Оформити документацію до дипломної роботи.

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

- Архітектура програмного забезпечення;
- Структурна схема програми.

6. Консультанти розділів проекту (роботи):

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	ст.викладач Мальчиков В.В.		

7. Дата видачі завдання «28» жовтня 2014



Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Вивчення літератури за тематикою роботи	13.11.14	
2.	Проведення порівняльного аналізу методів вирішення задач ціличисельного програмування	2.12.14	
3.	Вибір та обґрунтування методу вирішення задачі ціличисельного програмування	8.01.15	
4.	Проектування архітектури розроблюваних програмних засобів	15.01.15	
5.	Визначення складу та форматів вихідних даних та результатів програми	4.02.15	
6.	Розробка алгоритмів	18.02.15	
7.	Розробка мови управління програмами	18.03.15	
8.	Програмна реалізація	9.04.15	
9.	Розробка алгоритмів та підготовка контрольних задач для їх перевірки	23.04.15	
10.	Розв'язування контрольних задач на ПЕОМ	13.05.15	
11.	Оформлення документації дипломної роботи		

Студент _____ Вовченко О.О.

Керівник роботи _____ Олефір О.С.

АНОТАЦІЯ

Дана дипломна робота присвячена розробці інформаційно-довідкової системи підбору конфігурації комп'ютера.

В роботі розглянуті методи розв'язку задачі підбору конфігурації комп'ютера. Для розв'язку задачі було обрано перший алгоритм Гоморі.

Розроблено інформаційно-довідкову систему, яка використовує вищеописаний метод для визначення оптимальної конфігурації комп'ютера. Інформаційне забезпечення реалізовується шляхом використання СКБД.

Алгоритм реалізовано засобами мови програмування C#.

Проведено тестування та зроблені висновки на основі отриманих даних.

Ключові слова: КОНФІГУРАЦІЯ КОМП'ЮТЕРА, ІНФОРМАЦІЙНО-ДОВІДКОВА СИСТЕМА, МЕТОД ГОМОРІ



ABSTRACT

This thesis is devoted to developing informational system for selection configuration of the computer.

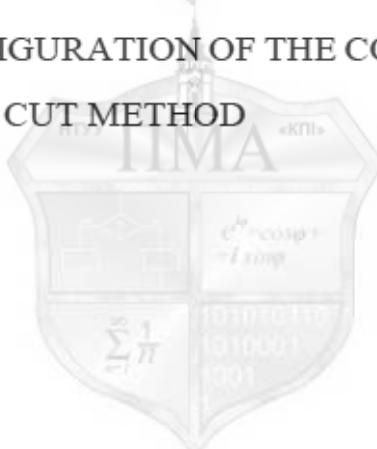
The paper discusses methods of solving problems selection of the configuration of the computer. For the problem solution was chosen Gomory's cut method.

The informational system, which uses the above method for selection configuration of the computer. Information support implemented by databases.

Algorithm is implemented using C#.

Testing of and conclusions based on the data obtained.

Keywords: CONFIGURATION OF THE COMPUTER, INFORMATIONAL SYSTEM, GOMORY'S CUT METHOD



Зміст

СПИСОК ТЕРМІНІВ, УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	11
ВСТУП	12
1 ПОСТАНОВКА ЗАДАЧІ.....	13
2 МАТЕМАТИЧНА МОДЕЛЬ ПОСТАВЛЕНОЇ ЗАДАЧІ	15
2.1 Математична модель задачі.....	17
2.2 Висновки.....	17
3 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ.....	18
3.1 Методи відтинання	18
3.1.1 Перший метод Гоморі.....	19
3.1.2 Другий метод Гоморі	20
3.1.3 Третій метод Гоморі.....	22
3.2 Метод віток та границь	25
3.2.1 Алгоритм Ленд-Дойг	27
3.3 Висновки.....	28
4 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	29
4.1 Категорії користувачів	29
4.2 Класи даних	30
4.3 Бізнес правила	30
4.4 Матриця елементарних подій	31
4.5 Інфологічне проєктування	32
4.5.1 Опис сутностей і атрибутів	33
4.5.2 Опис зв'язків	34

4.6 Даталогічне проектування	35
4.7 Розробка алгоритму	37
4.8 Висновки.....	38
5 КЕРІВНИЦТВО КОРИСТУВАЧА	39
5.1 Форма «Authorization».....	39
5.2 Форма «Registration»	40
5.3 Форма «Computer selection».....	42
5.4 «Information about computer»	45
5.5 «Optimized selection»	46
6. ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	48
6.1 Висновки.....	50
ВИСНОВКИ.....	51
ПЕРЕЛІК ПОСИЛАНЬ	52
Додаток А	53
Додаток Б	89

СПИСОК ТЕРМІНІВ, УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ЗЛП (Задача Лінійного Програмування) – задача оптимізації лінійної функції при лінійних обмеженнях.

ЗЦЛП (Задача Ціличисельного Лінійного Програмування) – задача оптимізації лінійної функції при лінійних обмеженнях, в якій на змінні накладено додаткову умову приймати лише ціличисельні значення.

СКБД (Система Керування Базою Даних) – комплекс програм, що забезпечує користувачам можливість створення, збереження, оновлення, пошук інформації та контролю доступу в базах даних.



ВСТУП

В останні роки галузь комп'ютерних технологій розвивається дедалі інтенсивніше, з кожним роком збільшуючи обчислювальні потужності настільних комп'ютерів збільшуються в півтора рази. Okрім цього збільшилась область використання комп'ютерів, наприклад з'явилися ігрові комп'ютери, комп'ютери для роботи з графічними редакторами і так далі.

Все це призводить до постійного зростання системних вимог програмного забезпечення, що в свою чергу спонукає споживачів оновлювати свою персональні комп'ютери якомога частіше.

У зв'язку з цим постає проблема розробки програмного засобу, який би надавав користувачу інформацію про комп'ютерні комплектуючі, які цікавлять його, та дозволив би клієнту зекономити час та гроші, автоматизувавши пошук оптимальної конфігурації комп'ютера, згідно тих характеристик та ціни, які потрібні користувачу.

1 ПОСТАНОВКА ЗАДАЧІ

Метою дипломної роботи є розробка довідково-інформаційної системи з підбору оптимальної конфігурації комп’ютера. Система повинна визначити найоптимальніший, з економічної точки зору, варіант набору комплектуючих комп’ютера користувача, згідно з його потребами. Також користувач повинен мати можливість переглянути інформацію про основні параметри вибраних комплектуючих, або обрати один з варіантів вже готових персональних комп’ютерів, які наявні в базі.

Для роботи з системою користувач повинен, спочатку зареєструватися, а вже потім працювати з системою. Це робиться для того, щоб для кожного клієнта зберігалася його історія переглядів. Після цього він повинен задати значення параметрів, які його цікавлять.

Головною задачею системи є економія часу клієнтів, який вони затрачують на пошук інформації, порівняння та вибір комплектуючих, поєднавши все це в одному програмному продукті.

Проблемною областью роботи є галузь контроля витрат.

Предметною областью є варіанти набору комплектуючих комп’ютера, які відповідають потребам користувача.

Об’єкт дослідження – покупці комп’ютерів.

Предмет дослідження – визначення конфігурації комп’ютера, яка задовольнятиме потреби клієнта, та буд найбільш вигідною в економічному плані.

До програмного забезпечення висуваються наступні вимоги:

- а) пошук оптимального товару ;
- б) Надання інформації про товар, який цікавить користувача;
- в) Здатність редагувати інформацію про товар;
- г) зручний інтерфейс користувача;
- д) формування історії користувача;

е) можливість подальшої модернізації.

До інформаційного забезпечення висуваються наступні вимоги:

- а) забезпечення цілісності даних;
- б) захист від несанкціонованого доступу;
- в) мінімізація введення та виведення інформації.



2 МАТЕМАТИЧНА МОДЕЛЬ ПОСТАВЛЕНОЇ ЗАДАЧІ

Метою даної роботи є забезпечити економію часу та коштів клієнта в процесі вибору нового персонального комп'ютера, або комп'ютера для підприємства. Для цього необхідно розробити математичну модель, результатом роботи якої буде набір комплектуючих, які найбільше підходять користувачу. Вибір здійснюється з деякої множини компонентів комп'ютера, які задовольняють вимогам клієнта, щодо їх продуктивності. Також вихідний набір комплектуючих не повинен містити елементів однакового типу.

Все це вказую нам на те, що поставлена задача є задачею цілочисельного лінійного програмування.



Цільова функція задачі лінійного цілочесельного програмування має наступний вигляд:

$$\min Z = f(\bar{X}, \bar{Y}, \bar{P}, \bar{K}) \quad (1)$$

Де \bar{X} – відеокарти, які задовольняють вимогам користувача,

\bar{Y} – підходящі процесори

\bar{P} – підходящі жорсткі диски

\bar{K} – підходяща оперативна пам'ять.

Тоді для задачі пошуку найбільш оптимальної за ціною конфігурації комп'ютера цільову функцію (1) можна записати наступним чином:

$$\min Z = \sum c_{x_i} x_i + \sum c_{y_j} y_j + \sum c_{p_l} p_l + \sum c_{k_r} k_r \quad (2)$$

де Z - ціна зібраного комп'ютера

c_{x_i} – ціна окремої відеокарти, $i = 1, \dots, n$;

x_i – відеокарта, $i = 1, \dots, n$;

c_{y_j} – ціна окремого процесора, $j = 1, \dots, m$;

y_j – процесор, $j = 1, \dots, m$;

c_{p_l} – ціна окремого жорсткого диску, $l = 1, \dots, d$;

p_l – жорсткий диск, $l = 1, \dots, d$;

c_{k_r} – ціна оперативної пам'яті, $r = 1, \dots, b$;

k_r – оперативна пам'ять, $r = 1, \dots, b$;

Причому

$$x_i = \begin{cases} 1 & \text{відеокарта вибрана} \\ 0 & \text{відеокарта не вибрана} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{процесор вибраний} \\ 0 & \text{процесор не вибраний} \end{cases}$$

$$p_l = \begin{cases} 1 & \text{жорсткий диск вибраний} \\ 0 & \text{жорсткий диск не вибраний} \end{cases}$$

$$k_r = \begin{cases} 1 & \text{оперативна пам'ять вибрана} \\ 0 & \text{оперативна пам'ять не вибрана} \end{cases}$$

На цільову функцію (2) накладаються наступні обмеження

$$\sum_{i=1}^n c_{x_i} x_i + \sum_{j=1}^m c_{y_j} y_j + \sum_{l=1}^d c_{p_l} p_l + \sum_{r=1}^b c_{k_r} k_r \leq S \quad (3)$$

$$\begin{cases} \sum_{i=1}^n x_i = 1 \\ \sum_{j=1}^m y_j = 1 \end{cases} \quad (4)$$

$$\begin{cases} \sum_{j=1}^m y_j = 1 \\ \sum_{l=1}^d p_l = 1 \end{cases} \quad (5)$$

$$\begin{cases} \sum_{l=1}^d p_l = 1 \\ \sum_{r=1}^b k_r = 1 \end{cases} \quad (6)$$

$$\begin{cases} \sum_{r=1}^b k_r = 1 \end{cases} \quad (7)$$

де S – максимальна ціна, яку готовий заплатити клієнт за новий комп’ютер.

Обмеження (3) гарантує, що сумарна вартість усіх обраних комплектуючих не перевищить ціни, яку готовий заплатити клієнт за новий комп’ютер.

Обмеження (4)-(7) гарантують вибір тільки однієї комплектуючої кожного типу.

2.2 Висновки

В даному розділі було сформовано математичну модель задачі оптимізації підбору конфігурації комп’ютера за ціною. Це виявилася задача цілочисельного лінійного програмування. Було прийнято рішення далі в цій роботі використовувати саме цю модель

3 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ

Оскільки, дана задача відноситься до задач цілочисельного лінійного програмування, то для її розв'язку будуть використовуватись методи для розв'язку задач цілочисельного програмування.

3.1 Методи відтинання

Ідея методів відтинання полягає ось у чому. Вирішується ЗЛП отримана з початкової ЗЦЛП шляхом відкидання умови цілочисельності. Переважно розв'язок не задовільнятиме умову цілочисельності. Тоді накладають додаткове обмеження, яке не виконується для отриманого плану задачі, проте задовільняє будь-який цілочисельний розв'язок. Таке додаткове обмеження називають правильним відтинанням. Система лінійних обмежень задачі доповнюється новою умовою і далі розв'язується отримана задача лінійного програмування. Якщо її розв'язок знову не задовільняє умови цілочисельності, то будується нове лінійне обмеження, що відтинає отриманий розв'язок, не зачіпаючи цілочислових планів. Процес приєднання додаткових обмежень повторюють доти, доки не буде знайдено цілочисельного оптимального плану, або доведено, що його не існує. [1]

3.1.1 Перший метод Гоморі

Перший метод Гоморі належить до розглянутого класу і полягає ось у чому. Розв'язується ЗЦЛП:

$$\begin{aligned} L(x) &= \sum_{j=1}^n c_j x_j \rightarrow \min \\ \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = 1, \dots, m \\ x_j &\geq 0, j = 1, \dots, n \end{aligned} \tag{1}$$

Розглядаємо допоміжну ЗЛП:

$$\begin{aligned} L(x) &= \sum_{j=1}^n c_j x_j \rightarrow \min \\ \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = 1, \dots, m \\ x_j &\geq 0, j = 1, \dots, n \end{aligned} \tag{2}$$

що одержується з (1) відкиданням умови ціличисельності змінних.

Нехай допоміжна ЗЛП (2) розв'язується симплекс-методом і на останній ітерації непрямі обмеження цієї задачі набули вигляду:

$$x_i + \sum_{j=m+1}^n a_{ij} x_j = \beta_i, \quad i = 1, \dots, m \tag{3}$$

i, таким чином, розв'язком допоміжної ЗЛП є n-вимірний вектор $\bar{x} = (\beta_1, \dots, \beta_m, 0, \dots, 0)$.

Отже, якщо до обмежень ЗЛП (3) або, що рівносильно, до непрямих обмежень ЗЛП (2), додати наступне обмеження

$$x_{n+1} - \sum_{j=m+1}^n \{a_{ij}\}x_j = \{\beta_i\}, \quad x_{n+1} \geq 0 \quad (4)$$

де x_{n+1} – додаткова змінна, і розв'язати ЗЛП (2), (4), то одержимо розв'язок, відмінний від $(\beta_1, \dots, \beta_m, 0, \dots, 0)$. Новий розв'язок також може бути неціличисельним, що приведе до необхідності додавання нового обмеження виду (4) і т. д. Зауважимо, що якщо в обмеженні (4) є індекс першої неціличисельної змінної, то можна гарантувати скінченність алгоритму першого методу Гоморі. [2]

3.1.2 Другий метод Гоморі

Другий метод Гоморі призначається для розв'язування частково (зокрема, повністю) ціличисельних ЗЛП:

$$L(x) = \sum_{j=1}^n c_j x_j \rightarrow \min \quad (5)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m \quad (6)$$

$$x_j \geq 0, j = 1, \dots, n \quad (7)$$

$$x_j - \text{ціле}, j = 1, \dots, p \quad (p \leq n) \quad (8)$$

Метод розв'язування задачі (5)–(8) ґрунтуються на тій же ідеї, що і метод розв'язування ЗЦЛП. А саме: розв'язується допоміжна ЗЛП (5)–(7), що одержується з вихідної відкиданням умови ціличисельності (8). Якщо ця задача розв'язку не має, то, очевидно, і вихідна частково ЗЦЛП не має розв'язку. Якщо ж ЗЛП (5)–(7) має розв'язок, то він аналізується на допустимість для задачі (5)–(8). Якщо знайдений оптимальний розв'язок є ціличисельним (у розумінні умов (8)), то він одночасно є оптимальним і для задачі (5)–(8). Інакше від розв'язаної ЗЛП переходят до нової допоміжної ЗЛП додаванням лінійного обмеження, яке задовольняють ціличисельні розв'язки вихідної часткової ЗЦЛП, але не задовольняє одержаний неціличисельний розв'язок вихідної ЗЛП. Це додаткове обмеження визначає деяку відтинаючу площину і називається правильним відтином. Додавання нових правильних відтинів відбувається доти, поки на деякому кроці не буде одержано ціличисельний розв'язок допоміжної задачі, що є, очевидно, оптимальним розв'язком часткової ЗЦЛП.

Нехай допоміжна ЗЛП розв'язується симплекс-методом і на останній ітерації непрямі обмеження цієї задачі набули вигляду:

$$x_i + \sum_{j=m+1}^n a_{ij} x_j = \beta_i, \quad i = 1, \dots, m \quad (9)$$

і, таким чином, розв'язком допоміжної ЗЛП є n -вимірний вектор $\bar{x} = (\beta_1, \dots, \beta_m, 0, \dots, 0)$.

Лінійне обмеження:

$$\sum_{j=m+1}^n \gamma_{rj} x_j \geq \{\beta_r\}$$

або, що рівносильно,

$$x_{n+1} - \sum_{j=m+1}^n \gamma_{rj} x_j = -\{\beta_r\}, x_{n+1} \geq 0 \quad (10)$$

де

$$\gamma_{rj} = \begin{cases} \{a_{rj}\} & \text{якщо } j \leq p, \{a_{rj}\} \leq \{\beta_r\} \\ \frac{\{\beta_r\}(1 - \{a_{rj}\})}{1 - \{\beta_r\}} & \text{якщо } j \leq p, \{a_{rj}\} > \{\beta_r\} \\ a_{rj} & \text{якщо } j > p, a_{rj} \geq 0 \\ \frac{\{\beta_r\}(-a_{rj})}{1 - \{\beta_r\}} & \text{якщо } j > p, a_{rj} < 0 \end{cases} \quad (11)$$

а x_{n+1} – додаткова змінна, є правильним відтином для частково ЗЦЛП (7) – (8).

Якщо r в обмеженні (10) є індекс першої нецілочисельної змінної серед перших p змінних, то алгоритм другого методу Гоморі є скінченим. [2]

3.1.3 Третій метод Гоморі

Розглядаємо ЗЦЛП:

$$L(x) = \sum_{j=1}^n c_j x_j \rightarrow \min \quad (12)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m \quad (13)$$

$$x_j \geq 0, j = 1, \dots, n \quad (14)$$

$$x_j - \text{ціле}, j = 1, \dots, n \quad (15)$$

Нехай непрямі обмеження ЗЛП (12)–(14), наприклад, у базисі A_1, \dots, A_m приведені до майже канонічного вигляду:

$$x_i + \sum_{j=m+1}^n a_{ij}x_j = \beta_i, i = 1, \dots, m \quad (16)$$

де $a_{ij}, \beta_i, i = 1, \dots, m, j = m + 1, \dots, n$, – цілі і симплекс-різниці $\Delta_j \geq 0, j = 1, \dots, n$. З непрямими обмеженнями (16) ЗЛП (12)–(14) називається майже канонічною. Якщо $\beta_i \geq 0, i = 1, \dots, m$, то обмеження визначають оптимальний розв'язок задачі (12)–(15), інакше визначається деякий ціличесельний майже допустимий базисний розв'язок вихідної ЗЛП. Можна було б, звичайно, вибрати один з індексів i , для якого $\beta_i < 0$, і виконати ітерацію двоїстого симплекс-методу. Проте у цьому випадку ціличесельність параметрів нової симплекс-таблиці була б, взагалі кажучи, порушена через необхідність ділення на ведучий елемент перетворення. Ціличесельність нової таблиці гарантується лише тоді, коли ведучий елемент дорівнює -1 .

Виявляється, що можна побудувати додаткове обмеження, якому задовольняють всі ціличесельні розв'язки задачі (12)–(15) і яке разом з тим визначає ведучий рядок перетворення, що має ведучий елемент -1 . Будується воно за l -м обмеженням системи (16), для якого $\beta_l < 0$:

$$x_l + \sum_{j=m+1}^n a_{lj}x_j = \beta_l \quad (17)$$

Якщо серед обмежень (16) є декілька з від'ємною правою частиною, то l вибирається, як правило, з умови $\beta_l = \min_{i: \beta_i < 0} \beta_i$.

Поділимо обидві частини (17) на довільне число $\alpha > 0$ і запишемо одержаний результат у вигляді:

$$\frac{x_l}{\alpha} + \sum_{j=m+1}^n \left\{ \frac{a_{lj}}{\alpha} \right\} x_j - \left\{ \frac{\beta_l}{\alpha} \right\} = \left[\frac{\beta_l}{\alpha} \right] - \sum_{j=m+1}^n \left[\frac{a_{lj}}{\alpha} \right] x_j \quad (18)$$

Ліва частина (18) при $x_j \geq 0, j = 1, \dots, n$, являє собою різницю двох величин

$$\frac{x_l}{\alpha} + \sum_{j=m+1}^n \left\{ \frac{a_{lj}}{\alpha} \right\} x_j \geq 0, \quad \left\{ \frac{\beta_l}{\alpha} \right\} < 1$$

тобто строго більша -1 . Отже, права частина, будучи цілим числом при ціличисельних x_j , задовольняє умову

$$\left[\frac{\beta_l}{\alpha} \right] - \sum_{j=m+1}^n \left[\frac{a_{lj}}{\alpha} \right] x_j \geq 0 \quad (19)$$

яка є вірною для будь-якого допустимого розв'язку задачі (12)–(15).

Уводячи додаткову змінну x_{n+1} , перепишемо (19) у вигляді

$$x_{n+1} + \sum_{j=m+1}^n \left[\frac{a_{lj}}{\alpha} \right] x_j = \left[\frac{\beta_l}{\alpha} \right], \quad x_{n+1} \geq 0 \quad (20)$$

Очевидно, що з від'ємності a_{lj} випливає від'ємність $\left[a_{lj}/\alpha \right]$ і навпаки. Тому $\left[\beta_l/\alpha \right] < 0$ і серед чисел $\left[a_{lj}/\alpha \right]$ є від'ємні, тобто рядок таблиці, що визначається новим обмеженням (20), може бути прийнятий за ведучий для наступного симплексного перетворення. Разом з тим при $\alpha = \max_{j=m+1, \dots, n} (-a_{lj})$

кожному від'ємному $a_{lj}, j = m + 1, \dots, n$, відповідає $[a_{lj}/\alpha] = -1$, тобто ведучий елемент цього перетворення явно дорівнює -1 .

Отже, розширюємо наявну симплекс-таблицю за рахунок $(m+1)$ -го рядка з елементами $[a_{lj}/\alpha]$ (елементи, що відповідають базисним змінним, рівні нулю) та одиничного стовпця A_{n+1} , що відповідає додатковій змінній x_{n+1} . Потім виконується симплекс-перетворення з $(m+1)$ -м ведучим рядком і ведучим стовпцем, що вибирається за правилами двоїстого симплекс-методу. Тоді нова симплекс-таблиця буде повністю ціличисельною. Описана послідовність дій складає окрему ітерацію алгоритму третього методу Гоморі. Ітерації виконуються доти, поки не буде отримана симплекс-таблиця, в якій усі праві частини невід'ємні, або є рядок з від'ємною правою частиною і невід'ємними рештою елементів. У першому випадку ЗЦЛП розв'язана, у другому — її обмеження є суперечливими. Якщо на будь-якій ітерації одна з додаткових змінних переходить з небазисних у базисні, то відповідні їй рядок і стовпець симплекс-таблиці викреслюються з подальшого розгляду. [2]

3.2 Метод віток та границь

Для розв'язання задач дискретного (зокрема ціличисельного) лінійного програмування широко використовується метод віток та границь. Цей метод належить до класу комбінаторних методів і зводиться до направленого перебору варіантів розв'язків оптимізаційної задачі, коли розглядаються лише ті з них, які виявляються за певними ознаками перспективними, і відкидаються відразу цілі множини варіантів, що є безперспективними.

Розглянемо загальну схему методу на прикладі оптимізаційної задачі

$$f(x) \rightarrow \min, x \in D \quad (21)$$

де D — скінчена множина, $x \in E^n$.

Основу методу складають такі процедури.

1. Обчислення нижньої оцінки (границі) для значень цільової функції $f(x)$ на допустимій множині $D = D^0$ (або на деякій її підмножині), тобто, знаходження числа $\xi(D^0)$, такого, що $f(x) \geq \xi(D^0)$ для всіх $x \in D^0$. Питання про те, як знаходиться $\xi(D^0)$, вирішується окремо для кожної задачі.

2. Розбиття на підмножини (розгалуження). Реалізація методу пов'язана з розгалуженням множини (або деякої її підмножини) в дерево підмножин згідно такої схеми.

Нульовий (початковий) крок. Деяким чином (в залежності від задачі) множина D^0 розбивається на скінченнє число підмножин $D^{1,1}, \dots, D^{1,N_1}$, таких, що не перетинаються між собою, і $D^0 = \bigcup_{l=1}^{N_1} D^{1,l}$.

На s -му кроці ($s \geq 1$) маємо множини $D^{s,1}, \dots, D^{s,N_s}$, одержані на попередньому кроці. За певним правилом (що формулюється нижче) серед них вибирається множина $D^{s,u}$, яка вважається перспективною. Ця множина розбивається на скінченнє число підмножин $D^{s,u,1}, \dots, D^{s,u,l}, \dots, D^{s,u,k_u}$ таких, що не перетинаються між собою, і $D^{s,u} = \bigcup_{l=1}^{k_u} D^{s,u,l}$.

Перепозначаємо множини $D^{s,1}, \dots, D^{s,u-1}, D^{s,u+1}, \dots, D^{s,N_s}$, що не розгалужувалися, та множини $D^{s,u,1}, \dots, D^{s,u,l}, \dots, D^{s,u,k_u}$, одержані розгалуженням $D^{s,u}$, через $D^{s+1,1}, \dots, D^{s+1,N_{s+1}}$.

3. Обчислення оцінок. На кожному кроці розгалуження знаходимо оцінки $\xi(D^{s,l})$, $l = 1, \dots, N_s$, такі, що $f(x) \geq \xi(D^{s,l})$ для всіх $x \in D^{s,l}$. В будь-якому випадку, якщо $D^{s,u} = \bigcup_{l=1}^{k_u} D^{s,u,l}$, то, як легко бачити, $\xi(D^{s,u,l}) \geq \xi(D^{s,u})$, $l = 1, \dots, k_u$.

4. Знаходження розв'язків. Для конкретних задач можна вказати різні способи (що, звичайно, визначаються специфікою задачі) знаходження допустимих розв'язків на послідовно розгалужуваних підмножинах.

5. Критерій оптимальності. Нехай $D^0 = \bigcup_{l=1}^{N_s} D^{s,l}$ і $x^{s,u} \in D^{s,u}$. Якщо при цьому $f(x^{s,u}) = \xi(D^{s,u}) \leq \xi(D^{s,l})$, $l = 1, \dots, N_s$, то $x^{s,u}$ є оптимальним розв'язком задачі (21). [2]

3.2.1 Алгоритм Ленд-Дойг

Далі наводиться алгоритм методу Ленд-Дойг, який являє собою реалізацію методу віток та границь для задачі цілочисельного лінійного програмування:

$$L(x) = \sum_{j=1}^n c_j x_j \rightarrow \min \quad (22)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m \quad (23)$$

$$x_j \geq 0, \quad j = 1, \dots, n \quad (24)$$

$$x_j \text{ — ціле}, \quad j = 1, \dots, n \quad (25)$$

де $R_i (i = 1, \dots, m)$ — будь-які з відношень $\leq, \geq, =$.

Деякі з чисел d_j в (24) можуть дорівнювати $+\infty$. Передбачається, що многогранна множина, яка визначена співвідношеннями (23), (24), є обмеженою.

Розв'язується допоміжна ЗЛП (22)–(24), яка отримана з вихідної ЦЗЛП (22)–(25) відкиданням умови цілочисельності змінних (25) (вітка 0;1). Якщо її розв'язок $x(0;1)$ — цілочисельний, то він же є і розв'язком вихідної ЗЦЛП. Інакше величина $\xi(0;1) = L(x(0;1))$ дає нижню оцінку (границю) цільової функції ЗЦЛП на множині $D(0;1) = D$, що визначається співвідношеннями (23)–(24).

Нехай деяка координата $x_j(0; 1)$ ($j = 1, \dots, n$) розв'язку $x(0; 1)$ не є ціличисельною. В цьому випадку здійснюється розгалуження множини $D(0; 1)$ на дві підмножини $D(1; 1)$ і $D(1; 2)$ додаванням до обмежень, що задають $D(0; 1)$, обмежень $x_j \leq [x_j(0; 1)]$ та $x_j \geq [x_j(0; 1)] + 1$ відповідно, де $[z]$ — ціла частина числа z . Далі розв'язуються нові допоміжні ЗЛП з обмеженнями, які визначаються підмножинами $D(1; 1)$ та $D(1; 2)$, знаходяться границі $\xi(1; 1)$ та $\xi(1; 2)$ і т. д.

Для подальшого розгалуження обирається перспективна множина $D(k; r)$ з найменшою границею $\xi(k; r)$. Процес продовжується доти, поки не буде отримано розв'язок, який задовольняє умову ціличисельності і для якого виконується ознака оптимальності. Внаслідок обмеженості допустимої множини ЗЛП (скінченності допустимої множини ЗЦЛП) метод Ленд-Дойг скінчений. [2]

3.3 Висновки



Під час вибору методу вирішення поставленої задачі перевага надавалась точним методам, бо наближені методи не гарантують вірної відповіді, що може бути критичним для дистрибутора.

Для поставленої задачі жоден з точних методів не проявив значних переваг над іншими, тому було обрано метод перший алгоритм Гоморі.

4 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Процес проектування програмного забезпечення складається з наступних етапів:

- а) інфологічне проектування;
- б) даталогічне проектування;
- в) розробка алгоритму.

Детальніше про кожен з цих етапів написано нижче.

4.1 Категорії користувачів



Користувачі даного програмного забезпечення розділяються на дві групи:

- а) клієнт;
- б) адміністратор.

Адміністратори можуть виконувати наступні дії:

- а) додавати та видаляти комплектуючі та комп’ютери;
- б) додавати, редагувати, видаляти характеристики комп’ютерів та їх комплектуючих;
- в) видаляти користувачів з системи;
- г) назначати нових адміністраторів;

Звичайні користувачі не мають доступу до зміни даних, за винятком своєї історії. Вони можуть лише переглядати інформацію про товар, який їх цікавить, та скористатися системою підбору конфігурації комп’ютера.

4.2 Класи даних

Для повноцінного функціонування підсистеми необхідні наступні вхідні дані:

а) дані про користувача:

- 1) логін;
- 2) пароль.

б) для пошуку оптимальної конфігурації :

- 1) значення характеристик комплектуючих;
- 2) максимальну ціну, яку клієнт готовий заплатити.

в) для пошуку інформації про комп'ютер:

- 1) назва комп'ютера;
- 2) назва комплектуючих;
- 3) тип комп'ютера;
- 4) діапазон ціни.

В результаті виконання операцій будуть отримані наступні дані про проект:

- а) Користувач зможе користуватися даним програмним засобом;
- б) Буду представлений оптимальний набір комплектуючих;
- в) Список підходящих комп'ютерів та інформація про них.

4.3 Бізнес правила

Програмне забезпечення відслідковує життєвий цикл наступних об'єктів:

- а) Компонента;

б) Комп'ютер;

Комплектуюча комп'ютера – з цих об'єктів складається кожний з комп'ютерів представлених в програмі. Створюється тоді, коли в продаж надходить нова компонента. Відповідно у систему заносяться відомості про її характеристики, модель та значення характеристик.

У разі вилучення компоненти з продажу відбувається каскадне видалення даних про неї. Спочатку видаляється інформація про її характеристики, а потім і сам об'єкт.

Комп'ютер – Створюється коли новий комп'ютер надходить у продаж. У системі зберігається інформація про його ціну, виробника, тип та список компонент з яких він складається.

Коли комп'ютер вилучається з продажу відбувається каскадне видалення інформації про нього. Спочатку видаляється інформація про те, з яких компонентів він складається, потім записи з історії користувача, а вже після цього інформація про сам комп'ютер.

4.4 Матриця елементарних подій

Матриця елементарних подій, яка відображає реакцію програмного забезпечення на різні дії користувача зображена в табл. 4.1

Таблиця 4.1 – Матриця елементарних подій

№	Опис події	Тип події	Реакція на подію
1	Користувач бажає визначити потрібні йому параметри	N	Надати форму для вводу параметрів
2	Користувач бажає редагуввати введені дані	N	Надати засоби редагування даних

Продовження таблиці 4.1

3	Реєстрація в програмі	N	Надати форму для реєстрації.
4	Адміністратор хоче видалити інформацію про комп'ютер	N	Надати вибір поля, зберегти зміни в БД.
5	Вхід користувача/адміністратора в систему	N	Надати форму для авторизації.
6	Був уведений невірний логін або пароль	N	Вивести повідомлення про невірні данні
7	Адміністратор хоче відредагувати дані	N	Надати вибір запису, зберегти зміни в БД.
8	Ввід некоректних даних	N	Очищення полів, видати повідомлення про помилку та повторне ведення.
9	Користувач залишив поля характеристик пустими	N	Вивести 100 перших комп'ютерів з бази
10	Втручання в роботу програми сторонніх осіб	NN	Передбачити захист інформації закріпленої за програмою.
11	Спроба вивести комп'ютер з ладу	NN	Передбачити захист інформації закріпленої за програмою.

4.5 Інфологічне проектування

Для побудови інфологічної моделі даних було використано модель типу «Сутність-зв'язок», яка складається з трьох конструктивних елементів: сутність, атрибут, зв'язок.

4.5.1 Опис сущностей і атрибутів

Основні сущності, а також їх атрибути, необхідні для створення оптимального підбору конфігурації комп'ютера:

- a) Computers – інформація про готові конфігурації комп'ютерів:
 - 1) comp_model_name – назва комп'ютера;
 - 2) price – ціна комп'ютера;
 - b) manufacturers – перелік виробників:
 - 1) manufacturer_name – назва виробника.
 - c) component_models – моделі комплектуючих:
 - 1) model_name – назва моделі.
 - g) component_types – типи компонентів які входять до складу комп'ютера:
 - 1) component_types – назва типу компоненти.
 - d) characteristics – характеристики, які мають конкретні типи компонент:
 - 1) char_name – назва характеристики.
 - e) users – список користувачів:
 - 1) nickname – логін користувача;
 - 2) password – пароль користувача.
 - f) user_categories – категорії користувачів:
 - 1) category – назва категорії;
- Додатково було створено проміжні сущності:
- a) computer_has_components – комплектуючі які входять до складу конкретного комп'ютера.
 - b) component_types_has_characteristics – вказує відношення між типом компонент та характеристиками.

в) model_characteristics_value – значення характеристик для конкретною моделі комплектуючої:

1) value – значення характеристики.

г) viewed_computers – історія переглядів користувача:

1) viewing_date – дата перегляду.

4.5.2 Опис зв'язків

Зв'язок між комп'ютером та його комплектуючою – «багато до багато», тобто у комп'ютера може бути декілька комплектуючих, а одна комплектуюча може входити в досить багато моделей комп'ютерів. Данна залежність виноситься в окрему таблиці.

Зв'язок між характеристикою та типом компоненти – «багато до багато», тобто кожен тип компонентів має певний набір характеристик, який притаманний тільки йому, однак деякі характеристики можуть повторюватися в різних типах комплектуючих. Ця залежність виноситься в окрему таблиці.

Зв'язок між моделлю комплектуючої комп'ютера та характеристиками конкретного типу комплектуючих – «багато до багато», тобто деяка модель комплектуючої має набір характеристик, який є унікальний для типу цієї комплектуючої, однак всі інші моделі того ж типу теж мають ці характеристики. Ця залежність виноситься в окрему таблицю.

Зв'язок між користувачем і переглянутими ним комп'ютерами – «багато до багато», тобто один комп'ютер можуть переглянути декілька користувачів, а один користувач може переглянути декілька комп'ютерів. Данна залежність виноситься в окрему таблицю.

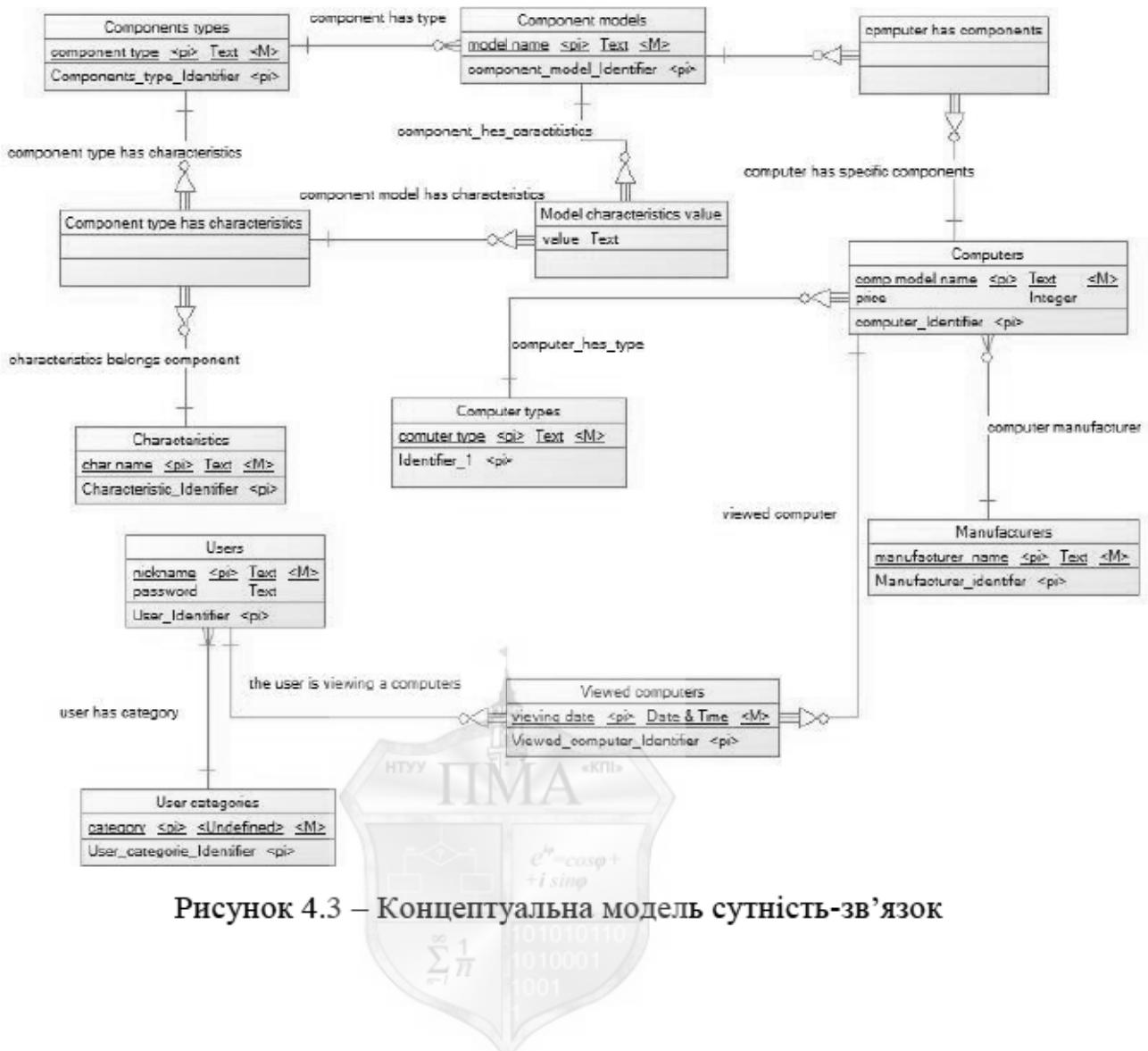


Рисунок 4.3 – Концептуальна модель сутність-зв’язок

4.6 Даталогічне проектування

На базі концептуальної моделі, описаної в розділі 4.6 , можна створити логічну модель. Для цього потрібно:

- для кожного атрибуту визначити тип;
- дляожної сущності визначити первинні ключі (PK);
- поєднати таблиці використовуючи зовнішні ключі (FK);
- зв’язки типу «багато до багатьох» замінити допоміжними таблицями.

До таблиць які є проміжними, при побудові зв’язків «багато до багато» додамо ключові атрибути тих сущностей, які зв’язують ці таблиці. Зробимо ці

атрибути ключовими. Результат виконання даних операцій можна побачити на рис. 4.4

Наступний крок – перехід до фізичної моделі даних. У фізичній моделі передбачається використання конкретної бази даних. В даній роботі використовується СКБД Oracle 10g. Архітектура програмного забезпечення «клієнтська». Уся логіка розміщена на клієнті. Результатуючу фізичну модель бази даних можна побачити на рис. 4.5

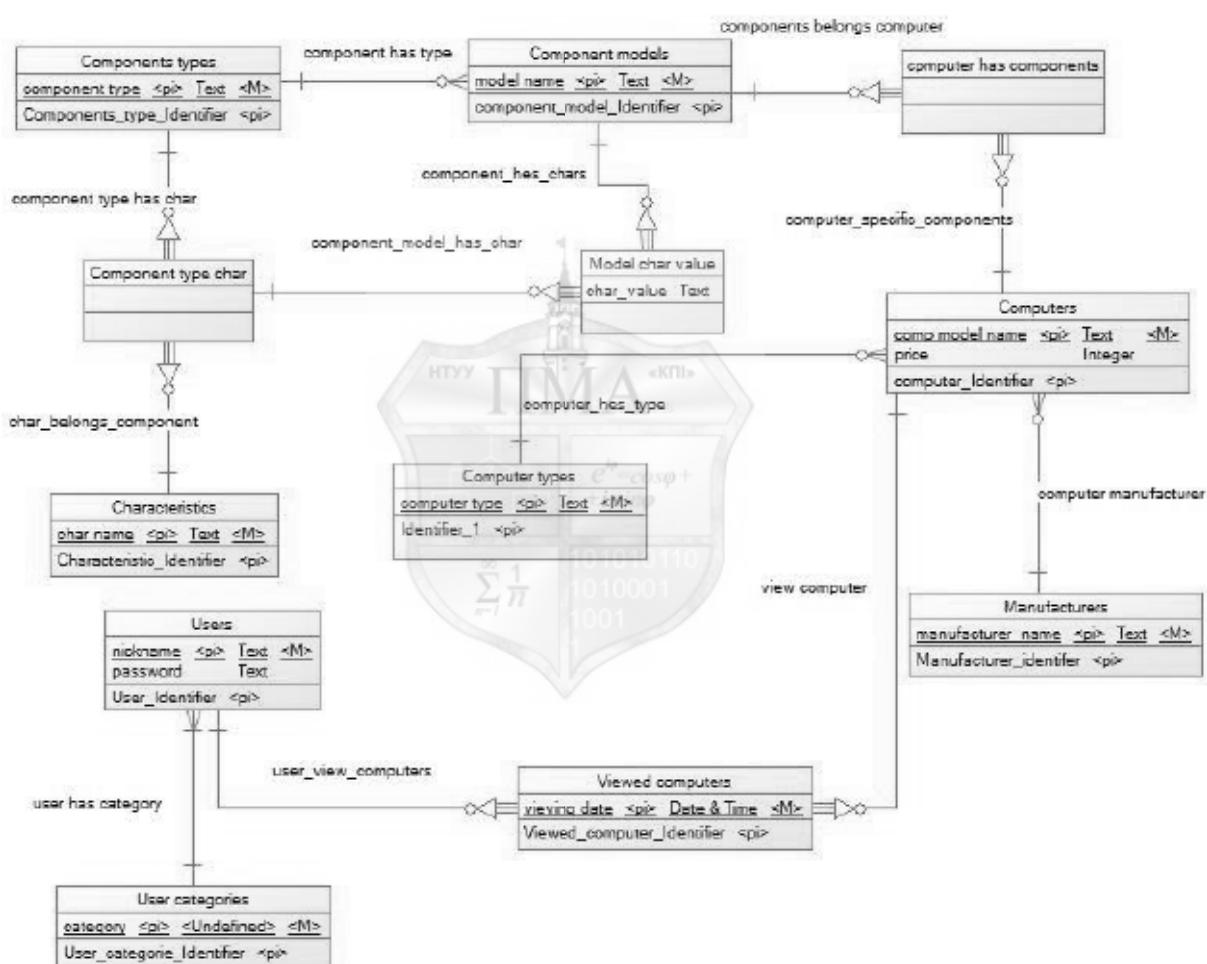


Рисунок 4.4 – Логічна модель

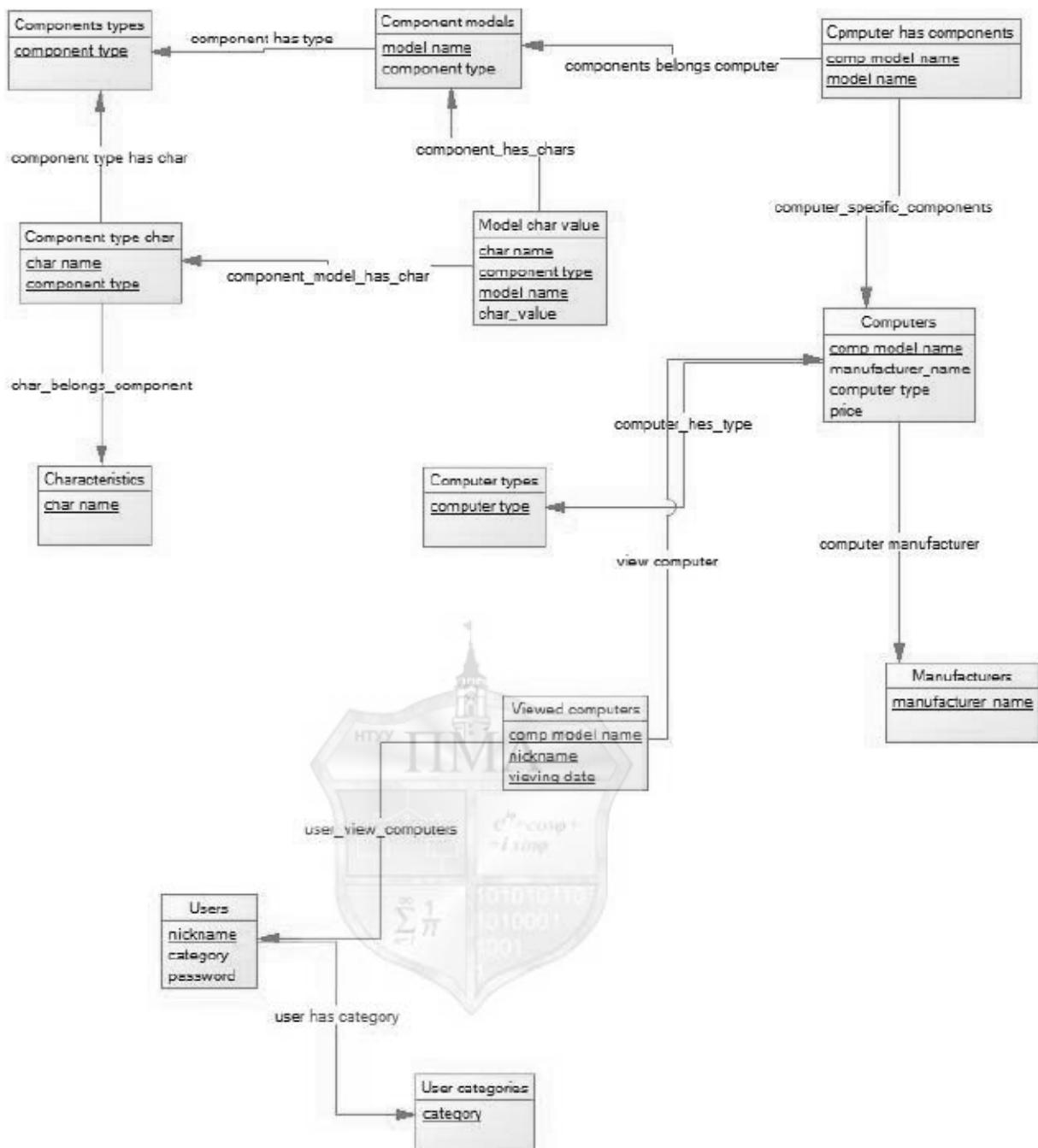


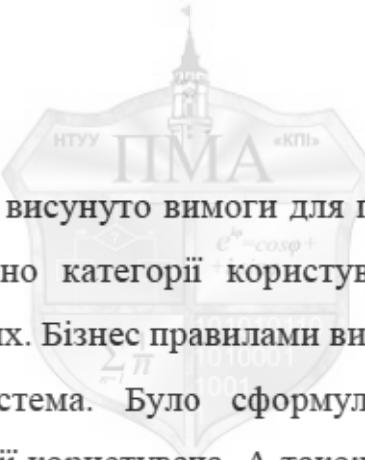
Рисунок 4.5 – Фізична модель сутність-зв’язок

4.7 Розробка алгоритму

Алгоритм Гоморі для повністю ціличисельних задач включає в себе наступні етапи:

1. Вирішується задача лінійного програмування, при цьому умова цілочисельності відкидається. Якщо отримана відповідь цілочисельна, то вона є рішенням. Якщо ні, то перехід на наступний крок.
2. Серед дробових чисел вибирається елемент з найбільшою дробовою частиною і складається додаткове обмеження.
3. Нерівність перетворюється в рівняння шляхом введення додаткової не від'ємної змінної.
4. Вирішується нова задача лінійного програмування. Якщо отримана відповідь цілочисельна, то вона є рішенням. Якщо ні, то перехід на крок

4.8 Висновки



В даному розділі висунуто вимоги для програмного та інформаційного забезпечення. Визначено категорії користувачів системи, а також класи вхідних і вихідних даних. Бізнес правилами визначено об'єкти, життєвий цикл яких відслідковує система. Було сформульовано реакцію програмного забезпечення на різні дії користувача. А також виконано моделювання бізнес процесів дипломної роботи. Проведено інфологічне та даталогічне моделювання.

5 КЕРІВНИЦТВО КОРИСТУВАЧА

В цьому розділі буде представлений огляд елементів інтерфейсу та їх реакції на дії користувача.

В даній програмі наявні наступні екранні форми:

- a) Authorization;
- б) Registration;
- в) Computer selection;
- г) Information about computer;
- е) Optimized selection.

Далі буде детальніше написано про кожну з них

5.1 Форма «Authorization»



На формі «Authorization»(мал. 5.1), яка буде показана клієнту першою, розташовані наступні елементи :

- а) кнопка «Enter»;
- б) кнопка «Registration»;
- в) поле для вводу «Enter your login»;
- г) поле для вводу «Enter your password»;
- е) пропорець «Show password».



Рисунок 5.1 – Форма «Authorization»

Перед тим як ввійти користуватися системою клієнт повинен авторизуватися. Для цього в поле «Enter your login» користувач повинен ввести свій логін, а в поле «Enter your password» пароль. Тільки після цього стане доступна кнопка «Enter». Якщо такий користувач зареєстрований в системі, то буде здійснено перехід на головну робочу форму «Computer selection»(рис. 5.3), в протилежному випадку буде надано можливість повторити спробу увійти в систему.

Для незареєстрованих користувачів передбачена кнопка «Registration», яка здійснює перехід до реєстраційної форми «Registration»(рис. 5.2).

5.2 Форма «Registration»

На реєстраційній формі «Registration» (рис. 5.2) розташовані наступні елементи :

- а) кнопка «Check in»;

- б) кнопка «Cancel»;
- в) поле для вводу «Enter your login»;
- г) поле для вводу «Enter your password»;
- д) поле для вводу «Retry password»;
- е) прапорець «Show password».



Рисунок 5.2 – Форма «Registration»

Для того щоб зареєструватися користувач повинен ввести логін, пароль та ще раз повторити свій пароль для підтвердження. Прапорець «Show password» дозволить перевірити правильність вводу паролю, якщо користувач забажає. Після того, як всі поля будуть заповнені стане доступна кнопка «Check in» яка запустить процедуру реєстрації. Якщо користувача з аналогічним логіном знайдено не буде, то данні клієнта будуть занесені в базу і буде повторно представлена форма авторизації для входу в систему, в протилежному випадку клієнту буде продемонстровано повідомлення про те, що такий логін вже зайнято та буде запропоновано повторно заповнити форму реєстрації.

Якщо з якихось причин користувач вирішить не реєструватися, то кнопка «Cancel» поверне його назад до форми авторизації.

5.3 Форма «Computer selection»

На формі «Computer selection»(рис. 5.3) знаходиться основна частина функціоналу. Вона здійснює пошук вже готових конфігурацій комп'ютерів за комплектуючими, які в них входять, надає користувачу переглянути свою історію та змінити пароль.

На першій вкладці(рис. 5.3) знаходяться наступні компоненти:

- а) пропорці для вибору компонентів, по яким буде здійснюватися пошук;
- б) списки компонентів;
- в) вкладка «User»;
- г) таблиця, в яку буде виводитись інформація про підходящі комп'ютери;
- д) кнопка «Find»;
- е) кнопка «Update»;
- ж) кнопка «Show more»;
- з) кнопка «Clean»;
- и) кнопка «Optimal configuration».

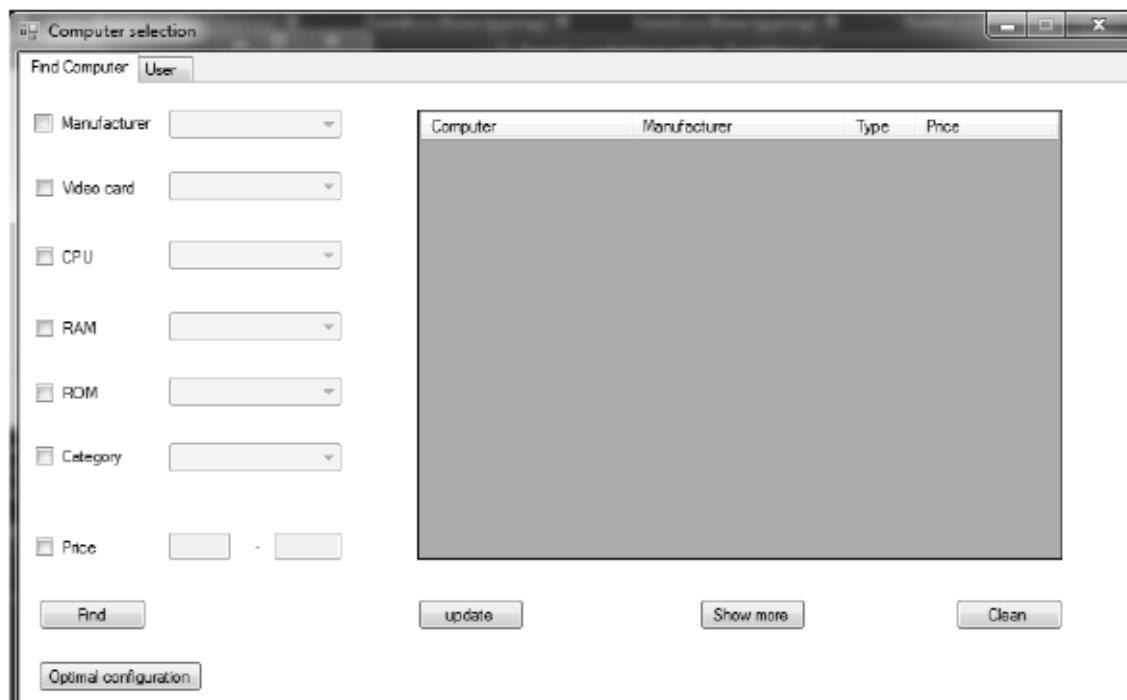


Рисунок 5.3 – Форма «Computer selection», вкладка «Find computer»

Для максимально ефективного пошуку користувач повинен встановити прaporці навпроти тих комплектуючих, які його цікавлять. Тоді після натискання кнопки «Find» в таблицю будуть виведенні всі ті комп’ютери, які задовольняють вибраним вимогам.

Якщо жоден прaporець не буде встановлений, тоді до таблиці будуть додані комп’ютери незалежно від їх характеристик.

Після того, як таблиця буде заповнена, користувач може подивитись інформацію про конкретний комп’ютер. Для цього йому потрібно вибрати один з них та натиснути кнопку «Show more», яка відкриє форму «Information about computer» (рис. 5.5).

Для того, щоб переглянути, чи не змінилося щось в базі треба натиснути кнопку «Update».

Для того щоб очистити таблицю необхідно натиснути кнопку «Clean».

Натискання кнопки «Optimal configuration» - дозволить скористатися функцією підбору оптимальної конфігурації комп’ютера відкривши форму «Optimized selection» (рис. 5.6).

На вкладці «User»(рис. 5.4) розташовані наступні компоненти:

- а) таблиця для відображення історії переглядів користувача;
- б) кнопка «Sign out»;
- в) кнопка «Change password»;
- г) кнопка «Confirm»;
- д) кнопка «Show history»
- е) поле вводу «Enter new Password»;
- ж) поле вводу «Retry password».

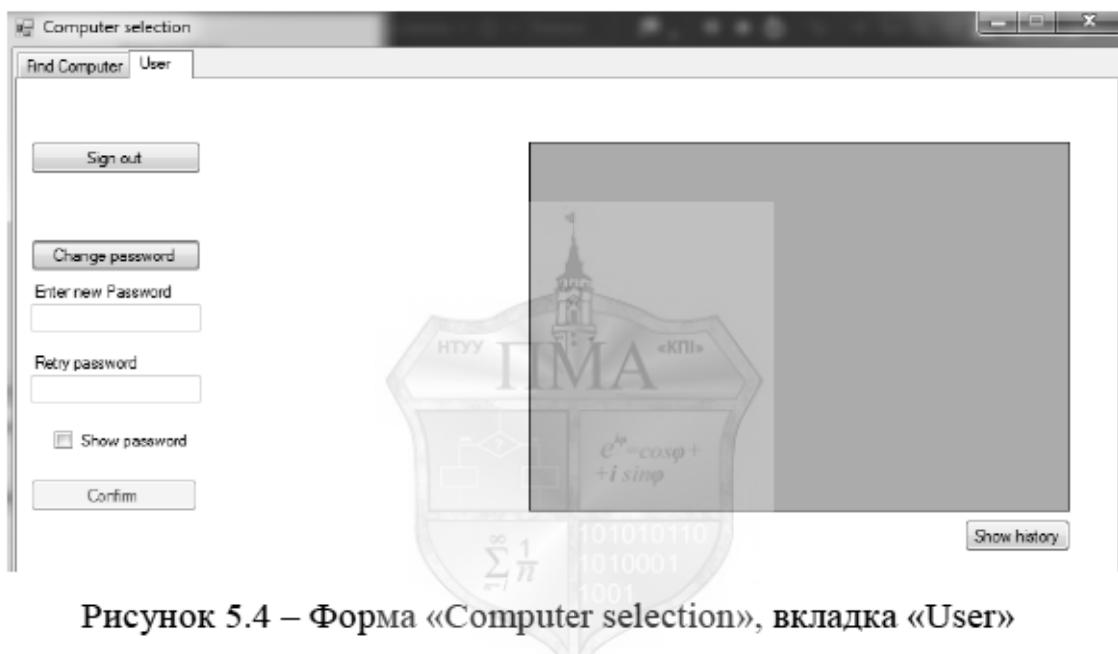


Рисунок 5.4 – Форма «Computer selection», вкладка «User»

На цій вкладці користувач може переглянути свою історію, для цього йому потрібно натиснути кнопку «Show history». Також є можливість вийти зі свого профілю натиснувши кнопку «Sign out», одразу після цього програма повернеться на форму «Authorization»(рис. 5.1).

Для того, щоб змінити пароль користувач повинен, спочатку натиснути кнопку «Change password», після цього відкриються поля «Enter new Password», «Retry password», в які він повинен ввести новий пароль, та кнопка «Confirm» яка стане доступна для натискання, тільки після заповнення цих полів.

5.4 «Information about computer»

На формі «Information about computer» (рис. 5.5) розташовані наступні елементи:

- а) поле «Model»;
- б) поле «Manufacturer»;
- в) поле «Type»;
- г) поле «Price»;
- д) таблиця для списку комплектуючих комп'ютера;
- е) таблиця для списку характеристик вибарної комплектуючої.

Component	Model
	$\sum_{n=1}^{\infty} \frac{1}{n^2}$
	1010110
	10001
	101

Characteristic	Value

Рисунок 5.5 – Форма «Information about computer»

В полях «Model», «Manufacturer», «Type», «Price» виводиться основна інформація про модель, виробника, тип та ціну комп'ютера. Таблиця списку комплектуючих надає інформацію про компоненти, з яких складається комп'ютер. Для того, щоб переглянути детальну інформацію про конкретну

деталь потрібно натиснути на неї і тоді в сусідній таблиці відобразиться список її характеристик та їх значень.

5.5 «Optimized selection»

На формі «Optimized selection»(рис. 5.6), яка відповідає за підбор найоптимальнішого варіанту конфігурації комп'ютера розташовані наступні елементи:

- а) поле вводу «Video card memory»;
- б) поле вводу «Processor Speed»;
- в) поле вводу «RAM memory»;
- г) поле вводу «ROM memory»;
- д) поле вводу «Max. Price»;
- е) випадаючий список «Type»;
- ж) кнопка «Selection»;
- ж) таблиця набору компонент.

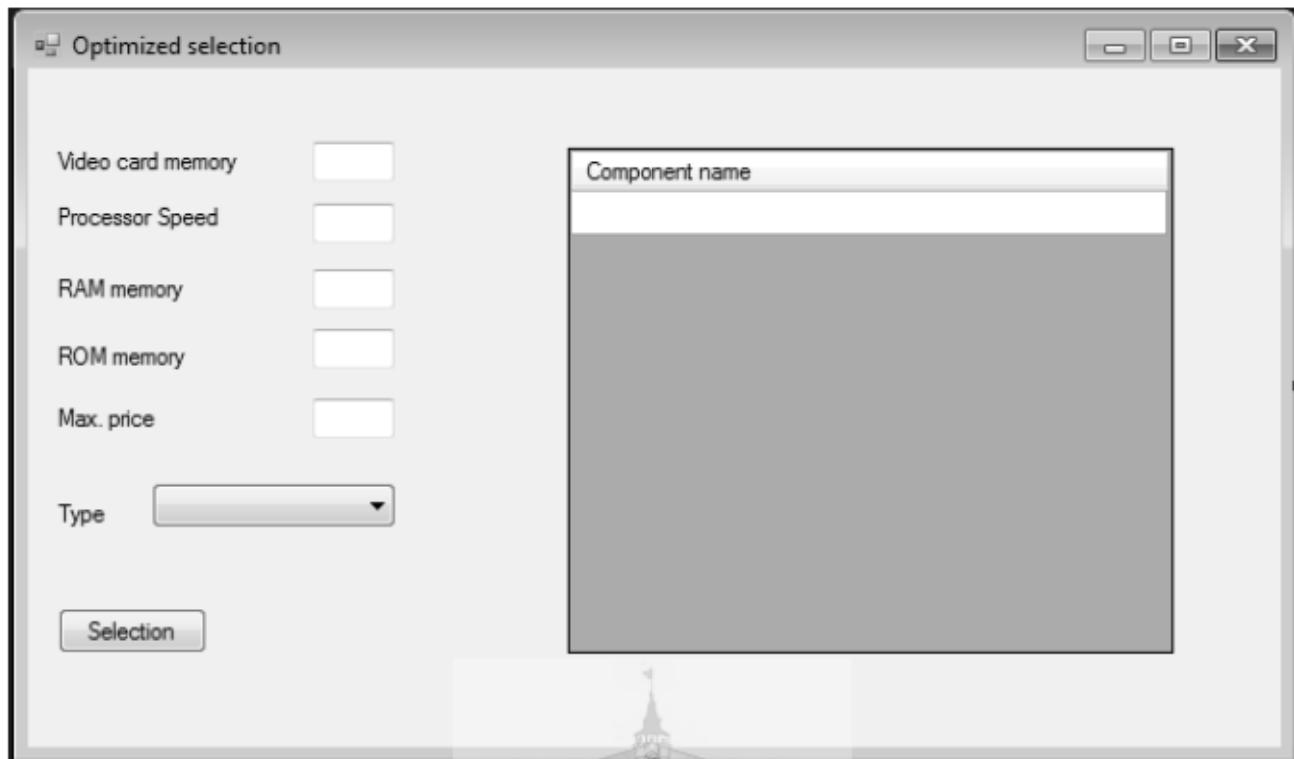


Рисунок 5.5 – Форма «Optimized selection»

Для максимально коректної роботи цієї функції користувачу рекомендується заповнити усі поля, але можливість оптимізації за ціною відкривається вже після заповнення одного з них.

Випливаючий список «Type» задає тип комп’ютера, напрклад ігровий. Інші поля, окрім ціни, задають мінімальні значення основних характеристик конкретних комплектуючих. Поле «Price» задає максимальну ціну підібраного комп’ютера. Якщо комп’ютера, який задовольняв би цій ціні не знайдено, то повідомлення про це буде виведено на екран.

Після натискання кнопки «Selection» буде розпочато процедуру підбору, після її завершення в таблиці будуть представлені назви компонентів, які складають оптимальну конфігурацію при заданих характеристиках та ціні.

6. ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тестування алгоритму полягає в перевірці правильності виконання оптимізації.

Програмне забезпечення було тестовано на простому прикладі, який можна порахувати вручну і перевірити правильність виконання.

Введемо наступні данні:

Таблиця 6.1 – Данні, які ввів користувач

Об'єм пам'яті відео карти	1 ГБ
Частота процесора	2.3 ГГЦ
Об'єм оперативної пам'яті	4 ГБ
Об'єм жорсткого диску	500 ГБ
Максимальна ціна	20000 грн

Після вибірки з бази даних підходящих елементів були вибрані комплектуючі, які розміщені в таблиці 6.2.

Таблиця 6.2 – Компоненти взяті з бази даних

Тип	Назва	Характеристика	Значення характеристики	Ціна
Відео карта	Palit PCI-Ex GeForce GTX 750 StormX	Об'єм пам'яті	1 ГБ	2679
Відео карта	Asus PCI-Ex Radeon R7 250	Об'єм пам'яті	1 ГБ	2275

Процесор	Intel Core i5-4690K	Частота процесора	3.5 ГГЦ	5956
----------	---------------------	-------------------	---------	------

Продовження таблиці 6.2

Процесор	AMD FX-9590	Частота процесора	4.7 ГГЦ	4840
ПЗУ	Western Digital Blue WD5000LPVX	Об'єм пам'яті	500 ГБ	1064
ПЗУ	Western Digital RE WD5003ABYZ	Об'єм пам'яті	500 ГБ	2016
ОЗУ	Goodram DDR2-800 4096MB	Об'єм пам'яті	4 ГБ	1848
ОЗУ	Kingston DDR3-1600 8192MB	Об'єм пам'яті	8 ГБ	1140

В результаті обчислень був отриманий наступний результат:

Таблиця 6.3 – Набір основних комплектуючих для нового комп'ютера

Тип комплектуючої	Назва	Ціна
Відео карта	Asus PCI-Ex Radeon R7 250	2275
Процесор	AMD FX-9590	4840
ПЗУ	Western Digital Blue WD5000LPVX	1064

ОЗУ	Kingston DDR3-1600 8192MB	1140
-----	------------------------------	------

Результатуюча ціна складає 9319 грн. що є значно менше зазначеної клієнтом.

Результат виконання програми із наведеними даними показано на рисунку (6.1):

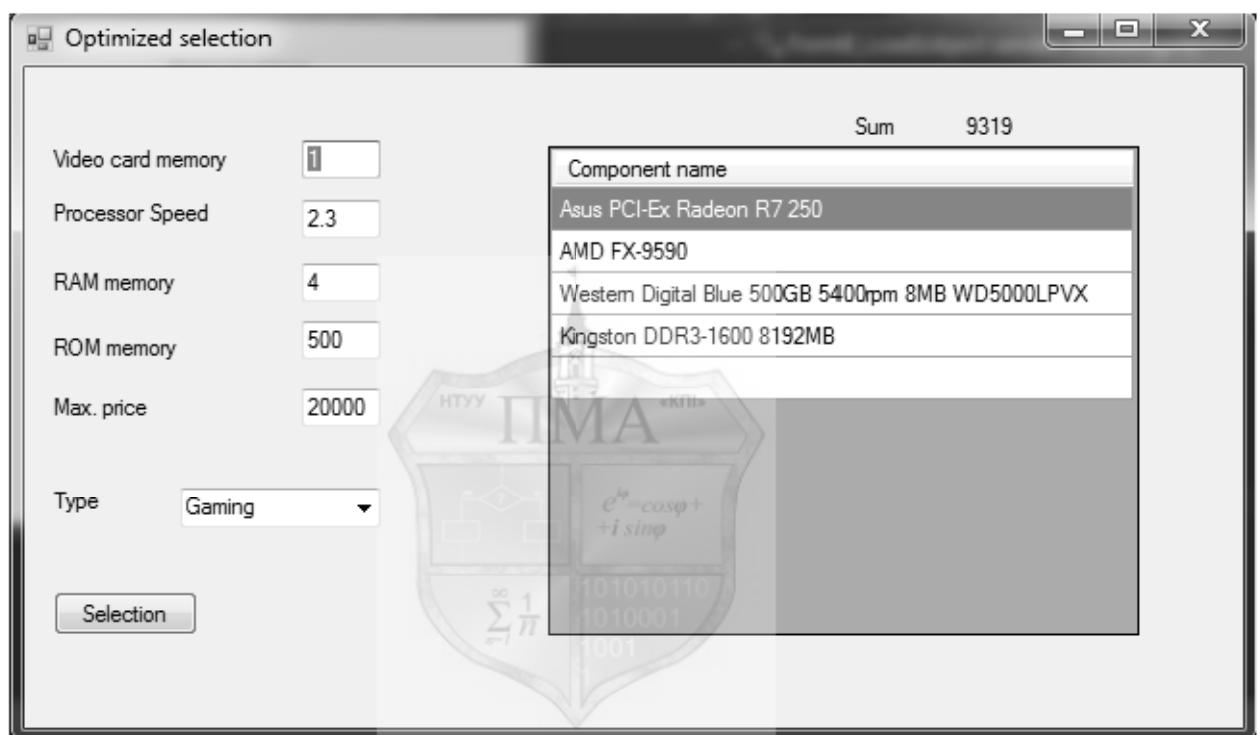


Рисунок 6.1 – Результат роботи програми

6.1 Висновки

В даному розділі було проведено тестування програмного забезпечення. Теоретичний результат повністю співпав із результатом отриманим програмним шляхом. З огляду на це можна стверджувати, що програмне забезпечення успішно пройшло тестування.

ВИСНОВКИ

В даній роботі, згідно з поставленою задачею, було розроблено програмне забезпечення для підбору конфігурації комп'ютера, яка була б оптимальною з економічної точки зору. Для цього було сформовано математичну модель поставленої задачі, проаналізовано різні методи розв'язку даної задачі, і вирішено використати перший метод Гоморі.

Програмне забезпечення було розроблено поетапно. Спочатку було виконано інфологічне та даталогічне проектування. Інформаційне забезпечення реалізовано СКБД Oracle 10g. Програмне забезпечення написане на мові програмування C#.

Було успішно проведено тестування програмного забезпечення.

ПЕРЕЛІК ПОСИЛАНЬ

1. Наконечний С. І., Савіна С. С. Математичне програмування: Навч. посіб. — К.: КНЕУ, 2003.
2. “Дискретне програмування”. Методичні вказівки до проведення практичних та самостійних занять з курсу “Дослідження операцій” для студентів факультету кібернетики / Упорядн. Володимир Іванович Тюптя, Віталій Іванович Шевченко, Віктор Кіндратович Стрюк. — К.: Електронне видання. Електронна бібліотека факультету кібернетики Київського національного університету імені Тараса Шевченка, 2003.

