

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ  
Навчально-науковий інститут денної освіти  
Форма навчання денна  
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту  
Завідувач кафедри  
\_\_\_\_\_ Олена ОЛЬХОВСЬКА  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

### **КВАЛІФІКАЦІЙНА РОБОТА**

**на тему**  
**«КОМБІНАТОРНІ ЗАДАЧІ ІГРОВОГО ТИПУ ТА МЕТОДИ ЇХ**  
**РОЗВ'ЯЗАННЯ»**

**зі спеціальності 122 Комп'ютерні науки**  
**освітня програма «Комп'ютерні науки»**  
**ступеня магістр**

**Виконавець роботи** Оріхівська Оксана Григорівна  
\_\_\_\_\_ « \_\_\_\_ » \_\_\_\_\_ 2024 р.  
(підпис)

**Науковий керівник** к.ф.-м.н., доц., Черненко Оксана Олексіївна  
\_\_\_\_\_ « \_\_\_\_ » \_\_\_\_\_ 2024 р.  
(підпис)

**Рецензент**

**ПОЛТАВА 2024 р.**

## **ЗМІСТ**

|                                                                                                                                                          |           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І</b>                                                                                                  | <b>3</b>  |
| <b>ТЕРМІНІВ</b>                                                                                                                                          |           |
| <b>ВСТУП</b>                                                                                                                                             | <b>5</b>  |
| <b>РОЗДІЛ 1. ОСНОВНІ ПОНЯТТЯ ТЕОРІЇ ІГОР ТА КОМБІНАТОРНОЇ</b>                                                                                            | <b>7</b>  |
| <b>ОПТИМІЗАЦІЇ</b>                                                                                                                                       |           |
| 1.1 Основні означення теорії ігор                                                                                                                        | 7         |
| 1.2 Властивості загальних множин переставлень та розміщень                                                                                               | 13        |
| 1.3 Задачі комбінаторної оптимізації на переставленнях та розміщеннях                                                                                    | 15        |
| 1.4 Задачі комбінаторної оптимізації ігрового типу на переставленнях та розміщеннях                                                                      | 17        |
| 1.5 Методи розв’язування задач комбінаторної оптимізації                                                                                                 | 22        |
| 1.6 Метод знаходження оптимальної стратегії гравця в матричних задачах теорії ігор з комбінаторними обмеженнями, які визначаються переставленнями        | 24        |
| <b>РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА</b>                                                                                                                      | <b>28</b> |
| 2.1 Ітераційний метод для знаходження оптимальної стратегії гравця, на стратегії якого накладаються комбінаторні обмеження, що визначаються розміщеннями | 28        |
| 2.2 Алгоритм методу                                                                                                                                      | 30        |
| 2.3. Аналіз часу роботи алгоритму                                                                                                                        | 31        |
| <b>РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА</b>                                                                                                                       | <b>37</b> |
| 3.1. Блок-схема алгоритму                                                                                                                                | 37        |
| 3.2. Опис програми                                                                                                                                       | 42        |
| 3.3. Результати розрахунків та їх аналіз                                                                                                                 | 46        |
| 3.4. Інструкція з використання розробленого застосунку.                                                                                                  | 50        |
| <b>ВИСНОВКИ</b>                                                                                                                                          | <b>52</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b>                                                                                                                        | <b>54</b> |
| <b>ДОДАТОК А</b>                                                                                                                                         | <b>56</b> |

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

| Позначення, термін    | Пояснення позначення, терміну                                                                                                                   |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| $A$                   | матриця значень функції виграшу                                                                                                                 |
| $a_{ij}$              | виграш першого гравця за рахунок другого в тому випадку, коли перший гравець застосує свою $i$ -ту стратегію, а другий – свою $j$ -ту стратегію |
| $[A]$                 | первинна специфікація (кортеж кратностей) мультимножини $A$ , упорядкована відповідно до елементів основи $S(A)$ мультимножини $A$              |
| $ A $                 | кількість елементів у множині чи мультимножині $A$                                                                                              |
| $A_n^k(G)$            | множина $k$ -розміщень без повторень з $n$ різних дійсних чисел                                                                                 |
| $\bar{A}_n^k(G)$      | множина $k$ -розміщень з повтореннями з $n$ різних дійсних чисел                                                                                |
| $A_{\eta}^k(G)$       | загальна множина $k$ -розміщень                                                                                                                 |
| $C$                   | коефіцієнт песимізму в критерії песимізму-оптимізму                                                                                             |
| $convM$               | опукла оболонка множини $M$                                                                                                                     |
| $E_{\eta}^k(G)$       | образ загальної множини $k$ -розміщень при її зануренні в арифметичний $k$ -вимірний евклідовий простір                                         |
| $e$ -задача           | евклідова задача комбінаторної оптимізації                                                                                                      |
| $e$ -множина          | евклідова комбінаторна множина                                                                                                                  |
| $F(X_i, Y_j)$         | функція прибутку в задачах комбінаторної оптимізації ігрового типу                                                                              |
| $F(X_{i_0}, Y_{j_0})$ | верхня ціна гри в задачах комбінаторної оптимізації ігрового типу                                                                               |
| $F(X_{i_0}, Y_{j_1})$ | нижня ціна гри в задачах комбінаторної оптимізації ігрового типу                                                                                |

|                                |                                                                                                    |
|--------------------------------|----------------------------------------------------------------------------------------------------|
|                                | типу                                                                                               |
| $G = \{g_1, g_2, \dots, g_n\}$ | мультимножина з елементами $g_1, g_2, \dots, g_n$                                                  |
| $J_n$                          | множина $n$ перших натуральних чисел                                                               |
| $J_n^0$                        | $J_n \cup \{0\}$                                                                                   |
| $k$ -вибірка                   | підмультимножина заданої мультимножини, яка містить $k$ елементів                                  |
| $k_A(a)$                       | кратність (число повторень) елемента $a$ у мультимножині $A$                                       |
| $P_k(G)$                       | множина переставлень без повторень з $k$ різних дійсних чисел з мультимножини $G$                  |
| $P_{kn}(G)$                    | множина переставлень з повтореннями з $k$ дійсних чисел, з яких $n$ різні                          |
| $P_{kn}^s(G, H)$               | поліпереставна множина                                                                             |
| $R^1$                          | множина дійсних чисел                                                                              |
| $R^k$                          | арифметичний $k$ -вимірний евклідовий простір                                                      |
| $r$                            | ранг системи лінійних нерівностей                                                                  |
| $S(A)$                         | основа мультимножини $A$                                                                           |
| $u$                            | параметр достовірності інформації про розподіл ймовірностей станів природи в критерії Ходжа-Лемана |
| $V(C) = v^1, \dots, v^l$       | фундаментальна система розв'язків системи лінійних нерівностей                                     |
| $X_{i_0}$                      | оптимальна стратегія першого гравця в задачах комбінаторної оптимізації ігрового типу              |
| $Y_{j_0}$                      | оптимальна стратегія другого гравця в задачах комбінаторної оптимізації ігрового типу              |
| $\Pi_{kn}(G)$                  | переставний многогранник                                                                           |
| $\alpha$                       | нижня ціна гри                                                                                     |
| $\beta$                        | верхня ціна гри                                                                                    |
| $\vartheta$                    | ціна гри                                                                                           |

## ВСТУП

Актуальним розділом методів досліджень та оптимізації операцій, котрий використовується у дослідженні ігрових задач, функціонуючих в умовах конфлікту є ігрові задачі для конфліктних випадків з комбінаторними обмеженнями стратегії на одного гравця[1-3]

Розвиток цього напрямку відбувся на основі прикладних проблем, котрі мають практичне використання в таких областях людської діяльності, як соціальні, економічні, екологічні та решта інших [4]. Основною властивістю цих задач є наступне: крім вже досліджених і відомих факторів виділяється ще й компонент невизначеності. До нього відносяться: невизначеність суперника – результат дій людини є непередбаченим через існування умов, вплив котрих змінює його, в тому числі через те, що передбачати наслідки дій інших людей, їх реакцію та вплив в повному обсязі не можливо; невизначеність природи – малодосліджені та недосліджені компоненти впливу, котрі змінюють протікання експерименту та впливають на результати ідентичних досліджень.

Спроби описати інформацію, котра перебуває під дією нестабільних чинників унеможливлені з математичної точки зору. Виконати такі описи в повному обсязі просто неможливо. Для того, щоб мати більш точне представлення про систему, котра перебуває під впливом нестабільних факторів, можна застосувати математичні засоби евклідової комбінаторної оптимізації.

Для того щоб поліпшити подібні моделі та перейти на якісно вищий рівень використовується теорія ігрових задач з комбінаторними обмеженнями на стратегії одного гравця. Специфіка цих задач дозволяє використовувати комбінаторні властивості множин для опису більш широкого класу задач.

*Метою роботи* є дослідження методів розв'язування комбінаторних оптимізацій задач ігрового типу, дослідження алгоритму розв'язування задач вказаного типу, а також розробка відповідного програмного забезпечення.

*Постає завдання* – дослідити моделі, алгоритми розв’язування комбінаторних задач ігрового типу, методів розв’язування даних задач і розробка програмного забезпечення для цих цілей.

*Об’єкт дослідження (розробки)* – комбінаторні задачі оптимізації, методи їх розв’язування та програмна реалізація.

*Предмет дослідження (розробки)* – дослідження та програмування методів розв’язування комбінаторних оптимізаційних задач ігрового типу, розробка програмного забезпечення для них.

*Основними методами* є теорія і методи евклідової комбінаторної оптимізації, алгебри, лінійного програмування та теорії ігор.

# РОЗДІЛ 1. ОСНОВНІ ПОНЯТТЯ ТЕОРІЇ ІГОР ТА КОМБІНАТОРНОЇ ОПТИМІЗАЦІЇ

## 1.1 Основні означення теорії ігор

Зачатки теорії ігор можна знайти ще в роботах Е. Бореля (1921 р.), але як самостійна дисципліна вона сформувалась значно пізніше, після публікації монографії Дж. Неймана та О. Моргенштерна "Теорія ігор і економічна поведінка" (1944 р.)

У сучасному суспільстві конфліктні ситуації трапляються дуже часто. Різні сторони конфлікту мають різні інтереси та використовують різні методи для досягнення своїх цілей. Ці протилежні інтереси і роблять ситуацію конфліктною. Для опису реальних конфліктних ситуацій використовується математичний апарат, який називається теорією ігор (ТІ) [5]. Щоб спростити аналіз конфліктних ситуацій і виключити несуттєві фактори, будуються спрощені моделі, які називаються *іграми*. Коли один з *гравців* обирає та виконує одну з дозволених правилами гри дій, це називається *ходом*. Існують різні типи ігор, котрі залежать від типу конфлікту.

В теорії ігор під *Стратегією* [1] розуміють набір правил, котрі однозначно вказують гравцю на вибір, що він має зробити під час кожного ходу в залежності від ситуації, що утворюється в результаті проведення гри.

*Оптимальною* вважається стратегія, котра забезпечує максимізований середній виграш при багаторазовому повторенні гри.

В залежності від характеру виграшів гри класифікуються як «ігри з нульовою сумою» та «ігри з ненульовою сумою».

Якщо в результаті гри сума виграшів всіх гравців залишається нулем гра називається *Грою з нульовою сумою*[6]. В результаті такої гри не відбувається ніякого притоку капіталу у гравців – виключно його розподілення між гравцями. Більшість ігор з економічною складовою є саме іграми з нульовою сумою – їх задача перерозподілити наявні капітали між присутніми гравцями, а не створити новий капітал. Якщо така гра відбувається між двома гравцями вона є *антагоністичною* – тобто протиставною. Оскільки виграш (збільшення

капіталу) одного гравця неможливий без програшу (зменшення капіталу) іншого.

Будь-який конфлікт, котрий може виникнути у людському суспільстві, можливо представити у вигляді скінченної чи нескінченної матричної гри. Для цього скінченна гра має відповідати наступним вимогам:

1. Взаємодія двох сторін є антагоністичною;
2. Кожна за сторін конфлікту має скінчену кількість можливих дій;
3. Дії однієї зі сторін конфлікту не впливають на дії іншої, і жодна зі сторін не володіє інформацією про дії іншої. Результати дій сторін оцінюються дійсним числом, котре визначає вигідність ситуації для однієї з них;
4. Обидві сторони оцінюють чи можуть їх дії змінити ситуацію на власну користь чи користь антагоніста;
5. Дії сторін конфлікту є неподільними і одноразовими, тобто структура дій не може мати якихось формальних властивостей, що відрізняються. Так можна дивитись на дії сторін як на елементи абстрактних множин, котрі спрямовані на поліпшення ситуації, що складається на власну користь.

Розглянемо поняття та терміни, котрі вводяться для подальшого викладення результатів дослідження.

*Означення 1.1* Матрична гра [6] – скінченна гра між двома гравцями, котра має нульову суму, де виграші першого гравця задаються у вигляді матриці. В цій матриці рядки використовуються як вказівник стратегій першого гравця, а стовпці як стратегії другого. Перетини рядків і стовпців заповнюються виграшами першого гравця (в тому числі від’ємними) в залежності від застосованих гравцями стратегій. Виграші другого гравця дорівнюють програшам, або від’ємним виграшам першого.

Матрична гра з нульовою сумою антагоністичного типу (двох гравців) може бути представлена у якості абстрактної гри двох гравців наступним чином:

Першому гравцю доступно  $m$  стратегій  $i = \overline{1, m}$ ; другий має в своєму розпорядженні  $n$  стратегій  $j = \overline{1, n}$ . Кожна пара стратегій  $(i, j)$  має відповідне



число  $a_{ij}$ , котре виражає виграш першого гравця відносно другого за умови, що перший гравець використовує стратегію  $i$ , в той час коли другий використовує стратегію  $j$ . Гравці роблять один хід: перший обирає свою  $i$ -ту стратегію ( $i = \overline{1, m}$ ), другий – свою  $j$ -ту стратегію ( $j = \overline{1, n}$ ), що призводить до виграшу першим гравцем значення, котре складає  $a_{ij}$  за рахунок другого гравця (коли виконується наступна умова « $a_{ij} < 0$ » перший гравець платить вказану суму  $a_{ij}$  другому).

Виграші записуються у прямокутну матрицю  $A$ :

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & \dots & n \end{matrix} & \downarrow \text{2-й гравець (стратегії)} \\ \begin{pmatrix} a_{11} & \dots & \dots & a_{1n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{m1} & \dots & \dots & a_{mn} \end{pmatrix} & \begin{matrix} 1 \leftarrow \text{1-й гравець (стратегії)} \\ 2 \\ \dots \\ m \end{matrix} \end{matrix}$$

Вона має назву *матриця виграшу*, або ж *платіжна матриця*.

Заповнення матриці вказує на завершення гри.

Розглядаючи матрицю  $A = (a_{ij})$ ,  $i = \overline{1, m}$ ,  $j = \overline{1, n}$ , можна стверджувати, що проведення кожної партії матричної гри з використанням матриці  $A$  зводиться до вибору першим гравцем  $i$ -го рядка;  $j$ -го стовпця другим гравцем і отримання першим гравцем вигоди, котра записується в матрицю  $A$  на перетині  $i$ -го рядка та  $j$ -го стовпця, навіть якщо вона від'ємна.

Кожну стратегію гравця  $i = \overline{1, m}$ ,  $j = \overline{1, n}$  часто називають чистою стратегією.

*Означення 1.2[6]* Число  $\alpha = \max_i \min_j a_{ij} = a_{i_0 j_0}$  називається «чиста нижня ціна гри». Воно демонструє який мінімальний виграш при використанні своїх чистих стратегій забезпечить собі перший гравець при всіх можливих діях, до котрих наблизиться другий гравець

*Означення 1.3* [6] Число  $\beta = \min_j \max_i a_{ij} = a_{i_1 j_1}$  називається чистою верхньою ціною гри. Дане число показує до яких максимальних програшів може призвести використання його стратегій гравцем два.

За умови, що в грі з матрицею  $A$  виконується рівність  $\alpha = \beta$ , тобто верхня чиста ціна збігається з нижньою чистою ціною, прийнято стверджувати, що така гра має сідлову точку в чистих стратегіях і має чисту ціну гри, що дорівнює:  $\vartheta = \alpha = \beta$

*Означення 1.4* Якщо гравець обирає всі доступні стратегії зі свого повного набору він використовує стратегію, котра називається мішана стратегія. Тобто, за умови, що гравець має  $m$  чистих стратегій  $i = \overline{1, m}$ , то його мішана стратегія  $x$  – це набір чисел  $x = (x_1, x_2, \dots, x_i, \dots, x_m)$ , що задовольняють умові  $x_i \geq 0, i = \overline{1, m}$ ,  $\sum_{i=1}^m x_i = 1$ .

Відповідно другий гравець, маючи  $n$  чистих стратегій  $j = \overline{1, n}$ , може використати мішану стратегію, де  $y$  – це набір чисел  $y = (y_1, y_2, \dots, y_j, \dots, y_n)$ , що задовольняють умовам  $y_j \geq 0, j = \overline{1, n}, \sum_{j=1}^n y_j = 1$ .

Чисті стратегії є несумісними подіями. Це пов'язано з тим, що при використанні гравцем однієї чистої стратегії він не зможе використати іншу, оскільки у нього не буде варіантів ходів, котрі він зможе використати.

*Означення 1.5* Середній виграш першого гравця в матричній грі з матрицею  $A$  може бути представлений у вигляді математичного сподівання його виграшів  $E(A, x, y) = \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_i y_j$ .

Особливим типом ігор вважається «гравець проти середовища». Така гра хоч і є антогоністичною, але на відміну від іншого гравця, середовище не зацікавлене в програші гравця. Воно зацікавлене у створенні умов чи слідуванню певному алгоритму визначених дій, що хоч і може призвести до програшу гравця, але не ставить собі за мету гарантування цього програшу. В залежності від того, яку ситуацію створює для гравця середовище і

співвідношення ризику до потенційного виграшу, гравцем можуть використовуватись наступні критерії для прийняття рішень:

1. *Критерій максимального математичного очікування* для потенційного виграшу використовується в випадках, коли стан середовища сприяє мінімізації шансу програшу, що дозволяє приділити більше уваги тому, щоб отримати більший виграш, хоч і в середньому. До платіжної матриці додається стовпець, кожний елемент котрого являє собою значення математичного сподівання виграшу при виборі відповідної  $i$ -ї стратегії:  $W_i = \sum_{j=1}^n a_{ij} p_j, i \in J_m$ . Оптимальною по даному критерію вважають ту стратегію, при виборі якої значення математичного очікування виграшу максимальне:  $W = \max_{i \in J_m} W_i$ .

2. *Критерій недостатньої основи Лапласа* прийнято використовувати для ситуацій, коли інформація про ймовірні стани природи не є вичерпною. Ці стани незначно відрізняються між собою і дозволяють мінімізувати ризики програшу, що дозволяє сконцентруватись на прийнятті рішень, котрі максимізують середній виграш. Шанс прийняття навколишнім середовищем

різними станами вважається рівним, тому  $W_i = \frac{\sum_{j=1}^n a_{ij}}{n}, i \in J_m$ . Оптимальною по даному критерію вважають ту стратегію, при виборі якої значення середнього виграшу максимальне:  $W = \max_{i \in J_m} W_i$ .

3. *Максимінний критерій Вальда* вважається доречним для використання в ситуаціях, коли можливість зміни чи появи станів природи не може бути передбачена і залишається невідомою. Задача може бути реалізована лише одноразово і треба мінімізувати, а краще повністю виключити, будь-які ризики. До платіжної матриці додається стовпець, котрий складений з елементів, що представляють собою мінімальне значення виграшу при виборі відповідної  $i$ -ї стратегії:  $W_i = \min_j a_{ij}, i \in J_m$ . Оптимальною по даному критерію вважають ту

стратегію, при виборі якої значення мінімальне значення виграшу максимальне:

$$W = \max_{i \in J_m} W_i.$$

4. *Критерій мінімаксного ризику Севіджа* є альтернативою використання максимінного критерію Вальда. Ключовою різницею є врахування ступеню впливу фактору, що може виникнути на величину потенційного виграшу. Створюється матриця ризиків  $R = (r_{ij}) = a_{\max j} - a_{ij}, i \in J_m, j \in J_n$ , котра доповнюється стовпцем, котрий призначений для запису максимального значення  $r_{ij}$  при виборі відповідної  $i$ -ї стратегії:  $R_i = \max_{j \in J_n} r_{ij}$ . Оптимальною по даному критерію вважається та стратегія, в якій значення  $R_j$  мінімальне:  $W = \min_{j \in J_n} R_j$ .

5. *Критерій песимізму-оптимізму (Гурвіца)* застосовується для ситуацій, коли інформація про стани середовища взагалі відсутня або вважається недостовірною. Треба враховувати імовірність виникнення будь-якого стану природи, реалізувати найменшу кількість розв'язків ситуації та допустити деякий ризик. З метою досягнення найбільш врівноваженої позиції вводиться оціночний коефіцієнт  $C$  – коефіцієнт песимізму, котрий перебуває в межах значень  $[0,1]$ . Він вводиться для відображення проміжних ситуацій між точкою зору граничного песимізму та граничного оптимізму. Якщо  $C=1$  даний критерій перетворюється на максимінний критерій Вальда. Якщо  $C=0$  він перетворюється на критерій «азартного гравця», котрий полягає у ставці на те, що випаде «найкращий випадок». Платіжна матриця отримує додатковий стовпець, котрий містить коефіцієнти, що обчислюються за формулою  $W_i = C \min_j a_{ij} + (1-C) \max_j a_{ij}$ . За цим критерієм оптимальною буде та стратегія, котра отримує максимальне значення  $W_i$  для  $W = \max_{i \in J_m} W_i$ .

6. *Критерій Ходжа-Лемана* призначений для використання в умовах, коли об'єм інформації про стани природи зібрана на основі невеликої кількості спостережень, є неточною і може змінюватись. Кожне рішення допускає нескінчену кількість реалізацій, а при невеликій кількості реалізацій допускається деякий ризик. Для оптимізації стратегії за цим критерієм

використовують параметр «достовірність інформації про стани природи». Він перебуває в межах  $[0,1]$ . При великому ступені достовірності вважається, що варто використовувати максимально математичне очікування виграшу, в іншому випадку - максимінний критерій Вальда. До платіжної матриці при використанні цього критерія додається стовпець, коефіцієнти якого обчислюються за наступною формулою:  $W_i = u \sum_{j=1}^n a_{ij} p_j + (1-u) \min_j a_{ij}$ . Оптимальна стратегія буде тією де значення  $W_i$  максимальне:  $W = \max_{i \in J_m} W_i$ .

## 1.2 Властивості загальних множин переставлень та розміщень

Задачі на переставних множинах відіграють важливу роль в комбінаторній оптимізації. Роботи Ю.Г. Стояна, О.О. Ємця та їх учнів [1-5, 7-10] присвячені дослідженню властивостей комбінаторних множин, занурених в арифметичний евклідовий простір, а також екстремальних властивостей опуклих функцій, заданих на комбінаторних множинах, та розробці методів розв'язання комбінаторних задач.

Припустимо, що  $J_n = \{1, \dots, n\}$ , а  $J_n^0 = J_n \cup \{0\}$ ,  $J_0 = 0$ , позначимо також  $\text{conv}M$  опуклу оболонку множини  $M$ .

Мультимножина - сукупність елементів, котрі можуть бути й однаковими. Будь-яка мультимножина  $A$  може бути представлена основою  $S(A)$ , тобто кортежем всіх різних елементів:  $S(A) = (a, b, \dots)$ , і кратністю – числом повторень кожного з елементів основи цієї мультимножини. Кратність елемента  $a \in S(A)$  записують як  $k_A(a)$ , кортежем кратностей  $(k_A(a), k_A(b), \dots)$  називається первинна специфікація. Її позначають  $[A]$ , причому елементи  $[A]$  впорядковані згідно з  $S(A)$ . Будемо позначати  $|A|$  кількість елементів у множині чи мультимножині  $A$ .

Мультимножина  $B$  з основою  $S(B)$  називається підмультимножиною мультимножини  $A$ , що має основу  $S(A)$ , якщо  $S(B) \subset S(A)$ , а для кожного з елементів  $a \in S(B)$  виконується нерівність  $k_B(a) \leq k_A(a)$  (позначається  $B \subset A$ ).

$k$ -вибіркою називається  $k$ -елементна підмультимножина мультимножини  $A$ , тобто  $B$  –  $k$ -вибірка, якщо  $B \subset A$ ,  $|B| = k$ .

Припустимо  $G = \{g_1, g_2, \dots, g_n\}$  – є мультимножиною з основою  $S(G) = (e_1, e_2, \dots, e_n)$ , де  $e_i \in R \ \forall i \in J_n$ , в якій кратності елементів  $k_G(e_i) = \eta_i$ ,  $\eta_1 + \eta_2 + \dots + \eta_n = \eta$ . Оскільки  $G$  – мультимножина, то можемо стверджувати, що існує принаймні одна  $i \in J_n$ , де  $k_G(e_i) = \eta_i > 1$ , а з цього випливає, що  $\eta > n$ .

Розглянемо упорядковану  $k$ -вибірку ( $k \in J_n$ )

$$e = (g_{i_1}, g_{i_2}, \dots, g_{i_k}) \quad (1.1)$$

з мультимножини  $G$ , де  $g_{i_j} \in G, i_j \neq i_t, \forall i_j, i_t \in J_n \ \forall j, t \in J_k$ .

Упорядковані  $k$ -вибірки  $e' = (e'_1, e'_2, \dots, e'_k)$  та з виглядом елементів  $e'' = (e''_1, e''_2, \dots, e''_k)$  називаються різними, якщо існує такий індекс  $j$  ( $j \in J_k$ ), що  $e'_j \neq e''_j$ .

*Означення 1.5* [7] Множина  $E$  називається евклідовою комбінаторною множиною, якщо її елементами є різні впорядковані  $k$ -вибірки вигляду (1.1).

Іншими словами, множина  $E$  має таку властивість: два елементи множини будуть відмінні один від одного, коли вони незалежно від інших відмінностей відрізняються порядком слідування символів, що їх утворюють.

Евклідову комбінаторну множину називають  $e$ -комбінаторною множиною [7], або коротко  $e$ -множиною. Розглянемо деякі  $e$ -комбінаторні множини.

Множина перестановок без повторень з  $k$  дійсних чисел – це множина, складена з упорядкованих  $k$ -вбірок вигляду (1.1) з мультимножини  $G$ , якщо  $\eta = n = k$ ,  $G = S(G)$  (мультимножина  $G$  є множиною). Такі  $k$ -вбірки можуть відрізнитись одна від одної тільки порядком слідування чисел і називаються переставленнями без повторень. Множина переставлень без повторень з  $k$  різних дійсних чисел позначається  $P_k(G)$ .

Приклад 1.

Нехай задана множина  $G = (a, b, c)$ . Сформуємо множину переставлень:  $P_k(G) = \{(a, b, c), (a, c, b), (c, b, a), (c, a, b), (b, a, c), (b, c, a)\}$ .

Множина  $k$ -розміщень без повторень з  $n$  різних дійсних чисел [7] – це множина всіх упорядкованих  $k$ -вбірок вигляду (1.1) з мультимножини  $G$ ,

якщо  $\eta = n$ ,  $G = S(G)$ . Цю множину розміщень позначають  $A_n^k(G)$ . Відображення  $f$ , яке кожному розміщенню множини  $A_n^k(G)$  ставить у відповідність точку арифметичного евклідового простору  $R^k$ , позначають  $E_n^k(G)$ , тобто  $E_n^k(G) = f(A_n^k(G))$ .

Приклад 2.

Нехай задана множина  $Z = (a, b, c)$ , сформуємо множину розміщень без повторень:  $A_3^2(Z) = \{(a, b), (b, a), (a, c), (c, a), (b, c), (c, b)\}$

Множина  $k$ -розміщень з повтореннями з  $n$  різних дійсних чисел [7] – це множина всіх упорядкованих  $k$ -вибірок вигляду (1.1) з мультимножини  $G$  за умови, що кратності  $\eta_i$  елементів  $e_i$  однакові та дорівнюють  $k \quad \forall i \in J_k$ , а  $\eta = nk$ . Позначимо цю множину розміщень  $\bar{A}_n^k(G)$ , а її образ  $\bar{E}_n^k(G) = f(\bar{A}_n^k(G))$ .

Приклад 2.

Нехай задана множина  $Z = (a, b, c)$ , сформуємо множину розміщень з повтореннями:  $A_3^2(Z) = \{(a, b), (b, a), (a, c), (c, a), (b, c), (c, b), (a, a), (b, b), (c, c)\}$ .

### 1.3 Задачі комбінаторної оптимізації на переставленнях та розміщеннях

Задачею комбінаторної оптимізації називають [2] задачу знаходження екстремуму

$$F(q^*) = \underset{q \in Q}{extr} F(q) \quad (1.7)$$

і екстремалі

$$q^* = \arg \underset{q \in Q}{extr} F(q), \quad (1.8)$$

де  $Q$  – скінченна множина комбінаторної природи,  $q$  – довільний елемент з  $Q$ ,  $F$  – деякий функціонал, заданий на  $Q$ . Розв'язком задачі (1.7), (1.8) називають пару  $\langle F(q^*), q^* \rangle$ .

$e$ -комбінаторна множина дозволяє виділити з множини задач комбінаторної оптимізації (1.7), (1.8) ті задачі, у яких множина  $Q$  є  $e$ -

множиною  $E$ , тобто задачі, що полягають у знаходженні пари  $\langle F(q^*), q^* \rangle$ , яка задовольняє умові:

$$F(g^*) = \underset{g \in E}{\text{extr}} F(g), \quad g^* = \arg \underset{g \in G}{\text{extr}} F(g). \quad (1.9)$$

Введення занурення  $f$  в арифметичний евклідовий простір  $R^k$  дає можливість замінити розв'язок задачі (1.9) розв'язком такої задачі: знайти

$$\Phi(y^*) = \underset{y \in E_f}{\text{extr}} \Phi(y), \quad y^* = \arg \underset{y \in E_f}{\text{extr}} \Phi(y), \quad (1.10)$$

де  $\Phi(y)$  – функція  $k$  змінних, визначена на множині  $E_f$ ,  $\Phi: E_f \rightarrow R_k$ , що відповідає функціоналу  $F(g)$ ,  $g \in E$ .

Задача (1.10) називається евклідовою задачею комбінаторної оптимізації або коротко  $e$ -задачею [7].

Часто зручно задачу (1.10) зобразити в такому вигляді: знайти

$$H(y^*) = \underset{x \in E_{\phi f}}{\text{extr}} H(y), \quad y^* = \arg \underset{x \in E_{\phi f}}{\text{extr}} H(y), \quad (1.11)$$

при обмеженнях

$$\phi^i(y) \leq 0 \quad \forall i \in J_r, \quad (1.12)$$

$$\phi^{r+i}(y) = 0 \quad \forall i \in J_s, \quad (1.13)$$

де  $x = (x_1, \dots, x_k)$ ,  $y = (x_1, \dots, x_k, y_{k+1}, \dots, y_m) \in R^m$ ,  $m$  – деяка натуральна константа ( $m \geq k$ ), а  $H(y)$ ,  $\phi^i(y) \quad \forall i \in J_{r+s}$  – функції  $m$  змінних. Обмеження (1.12), (1.13) називають додатковими обмеженнями  $e$ -задачі, а  $x \in E_{\phi}$  – комбінаторним обмеженням.

Тим часом в залежності від вигляду множини можливо розглянути задачу безумовної оптимізації для загальної множини перестановок: знайти пару  $\langle F(x^*), x^* \rangle$  таку, що  $F(x^*) = \underset{x \in E_{kn}(G)}{\text{extr}} F(x)$ ,  $x^* = \arg \underset{x \in E_{kn}(G)}{\text{extr}} F(x)$ , та задачу безумовної лінійної оптимізації на загальній множині розміщень : знайти пару  $\langle F(x^*), x^* \rangle$  таку, що  $F(x^*) = \underset{x \in E_{\eta m}^k(G)}{\text{extr}} F(x)$ ,  $x^* = \arg \underset{x \in E_{\eta m}^k(G)}{\text{extr}} F(x)$ .



#### 1.4 Задачі комбінаторної оптимізації ігрового типу на переставленнях та розміщеннях

В теорії ігор окремим класом задач є так звані “ігри з природою”. В них невизначеність результату залежить від об’єктивної дійсності, котру прийнято називати природою. Другим гравцем в таких іграх є природа (погода, попит на продукцію і т.д.). Розглянемо наступну задачу.

*Задача 1* (сільськогосподарського виробництва) [9].

Фермерське господарство володіє кількістю полів, що дорівнює  $m$  - кожне поле з різною площею. На цих полях фермерство вирощує  $m$  різних видів сільськогосподарських культур. Сумарний врожай культури залежить від вибору поля, на котрому вона буде висаджена. Також, треба мати на увазі, що кожна культура має власну врожайність, що залежить від природних факторів (переважно погоди). Метою є пошук оптимального плану для вирощування культур, щоб максимізувати прибуток.

Ця задача має ознаки задачі з комбінаторними обмеженнями на стратегії першого гравця, а антагоністом є природа. Для “ігор з природою” в класичній теорії ігор розроблені принципи, котрі допомагають перетворити невизначені ситуації на визначені. Володіючи інформацією про стан природи, кількість реалізацій ситуації та відношення людини, відповідальної за прийняття рішення, необхідно обрати один з усіх розглянутих критеріїв, який забезпечить найбільш оптимальний результат.

Позначимо  $P_i^x$  – відношення площі  $i$ -го поля до суми площ всіх полів господарства. Тоді вектор  $P^x = (P_1^x, P_2^x, \dots, P_m^x)$  показує, яку частину від загальної суми площі полів складає кожна ділянка. Якщо це частина від загальної площі то

$$\forall i \in J_m \quad P_i^x \geq 0; \quad \sum_{i=1}^m P_i^x = 1.$$

Позначимо вектор  $X = (x_1, x_2, \dots, x_m)$ , що характеризує загальну кількість вирощеної продукції господарством, відповідно його компонента за номером  $i$

є відношенням кількості вирощеної продукції сорту  $i$  до кількості продукції, що могла бути вирощеною, якби її вирощували на всіх полях.

Так як кожна з культур вирощуються тільки на одному з полів, то  $i$ -а компонента вектора  $X$  чисельно буде дорівнювати тій компоненті вектора  $P^x$ , яка відповідає полю, на якому вирощено культуру типу  $i$ . Тобто компоненти вектора  $X$  будуть отримуватися, як переставлення компонент вектора  $P^x$

Тоді

$$X = (x_1, x_2, \dots, x_m) \in E_m(P^x);$$

де  $E_m(P^x)$  – множина переставлень з елементів вектора  $P^x$ .

Розглянемо іншу задачу планування виробництва, яка буде відрізнятися від розглянутих вище задач тим, що тепер на стратегію першого гравця антагоністичної гри будуть покладені комбінаторні обмеження, що визначенні розміщенням.

*Задача 2* (задача промислового виробництва) [10].

Припустимо, що в місті знаходяться дві конкуруючі організації, котрі займаються виробленням та збутом хлібобулочних виробів. Перше підприємство виготовляє за зміну  $\gamma$  продукту. Хліб, як продукт завжди користується попитом і швидко реалізується (вважаємо, що вчорашній хліб не підлягає реалізації), що створює для підприємств проблему з щоденним розвезенням продукції. Для того, щоб розвести у  $m$  фірмових магазинів продукцію першому підприємству потрібно  $m$  автомобілів, котрі використовується для доставки хлібобулочних виробів щоранку. Автопарк підприємства складається з  $M$  автомобілів ( $M > m$ ) різної вантажопідйомності  $\xi$ . Для другого підприємства не існує проблем з доставкою хліба, тому нас цікавить тільки кількість магазинів, що дорівнює  $n$ . Залежно від кількості хліба, що буде завезена у кожний фірмовий магазин формується прибуток обох хлібозаводів.

Нехай  $P_i^x$  – відношення кількості продукції, яку можна відвезти  $i$ -им із запропонованих першому підприємству автомобілів, до всієї кількості

продукції, яку виробляє перший виробник  $\frac{\gamma}{\xi}$ . Відповідний вектор

$P^x = (P_1^x, P_2^x, \dots, P_M^x)$ , для елементів якого буде виконуватися:

$$P_i^x \geq 0 \quad \forall i \in J_M.$$

Покладемо  $X = (x_1, x_2, \dots, x_m)$  – вектор, що вказує на обсяги розвезення продукції першим підприємством. Його складова з номером  $i$  – це частина продукції першого підприємства, що була завезена в  $i$ -ий магазин. Продукція в кожний магазин привозиться одним автомобілем, а автомобілі вибираються з  $M$  доступних для використання, тоді:

$$X = (x_1, x_2, \dots, x_m) \in E_M^m(P), \quad (1.14)$$

де  $E_M^k(P^x)$  – множина  $k$ -розміщень з  $M$  елементів вектора  $P^x$ .

Обов'язковою умовою є розвезення всієї продукції в магазини, виходячи з цього компоненти вектора  $X = (x_1, x_2, \dots, x_m)$  повинні задовольняти наступну

умову:

$$\sum_{i=1}^m x_i = 1;$$

$$\tilde{o}_m = P_i^x \leq 1 - \sum_{i=1}^{m-1} x_i \quad \exists P_i^x \in P$$

$(x_1, \dots, x_m)$  – є допустимим розв'язком, тоді коли існує  $P_i \in P$ , тобто всі автомобілі, крім останнього, будуть розвозити частину продукції  $x_i, i \in J_{m-1}$ , а останній автомобіль – частину продукції  $1 - \sum_{i=1}^{m-1} x_i$ . Тому замість умови (1.14) доцільніше

позначити  $\bar{X} = (x_1, x_2, \dots, x_{m-1})$  та розглядати таку умову:

$$\bar{X} = (x_1, x_2, \dots, x_{m-1}) \in E_M^{m-1}(P^x).$$

Нехай друге підприємство (другий гравець) має  $n$  чистих стратегій.

Складемо відповідну матрицю  $A$  вигляду:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m1} & \dots & a_{mn} \end{pmatrix}. \quad (1.15)$$

Елемент  $a_{zj}$  показує різницю між прибутками першого і другого підприємств тоді, як вся продукція першого відвозилась у відповідний магазин з номером  $^3$ , а вся продукція другого – в його  $j$ -ий магазин.

Вектор  $Y = (y_1, y_2, \dots, y_n)$  – мішана стратегія другого підприємства – набір ймовірностей використання кожної чистої стратегії. Оскільки  $y_j$  – ймовірності повної групи подій, тоді

$$\forall j \in J_n \quad y_j \geq 0; \quad \sum_{j=1}^n y_j = 1.$$

Позначимо функцію прибутку як  $F(X_i, Y_j)$ . Середній прибуток першого підприємства, котрий отримується при використанні  $j$ -ї чистої стратегії другого гравця, дорівнює  $\sum_{i=1}^m a_{ij} x_i$ , а тому при застосуванні змішаної стратегії  $Y = (y_1, y_2, \dots, y_n)$  прибуток буде визначатися математичним сподіванням

$$F(X_i, Y_j) = \sum_{j=1}^n y_j \sum_{i=1}^m a_{ij} x_i.$$

Поведінку гравців можемо описати таким чином: перший гравець – за рахунок зміни компонент вектора  $X$  прагне максимально збільшити свій мінімальний прибуток  $F(X_i, Y_j)$  отже треба знайти такі  $X$  та  $Y$ , за яких досягається нижня ціна гри:  $\alpha = \max_{i \in J_m} \min_{j \in J_n} F(X_i, Y_j)$ ; а другий гравець, в свою чергу, за рахунок своїх мішаних стратегій прагне зробити свій максимальний програш мінімальним – верхню ціну гри:  $\beta = \min_{j \in J_n} \max_{i \in J_m} F(X_i, Y_j)$ .

Отримано наступну математичну модель задачі:

Знайти оптимальні стратегії гравців  $X_{i_0}$  і  $Y_{j_0}$ , де

- $X_{i_0}$  знаходиться з виразу  $F(X_{i_0}, Y_{j_1}) = \max_i \min_j F(X_i, Y_j)$ ;
- $Y_{j_0}$  – з виразу  $F(X_{i_1}, Y_{j_0}) = \min_j \max_i F(X_i, Y_j)$ ;
- функція  $F(X_i, Y_j)$  має вигляд  $F(X_i, Y_j) = \sum_{j=1}^n y_j \sum_{i=1}^m a_{ij} x_i$ ;

- $a_{ij} (\forall i \in J_m \quad \forall j \in J_n)$  – задані числа;

при обмеженнях:

$$1. \quad \bar{X} = (x_1, x_2, \dots, x_{m-1}) \in E_M^{m-1}(P^x); \quad X = (x_1, \dots, x_m): \sum_{i=1}^m x_i = 1;$$

вектор  $P^x = (P_1^x, P_2^x, \dots, P_M^x)$  задовольняє умовам  $P_i^x \geq 0 \quad \forall i \in J_M$ ;

$$2. \quad Y = (y_1, y_2, \dots, y_n) \text{ задовольняє умовам } \forall j \in J_n \quad y_j \geq 0; \quad \sum_{j=1}^n y_j = 1.$$

Задача, що вище описана є задачею комбінаторної оптимізації ігрового типу з комбінаторними обмеженнями на множині розміщень на стратегії одного гравця.

### Задача 3.

Розглянемо задачу 2.

Припустимо, що на території першого підприємства знаходиться його фірмовий магазин, тоді не вся продукція вивозиться з хлібозаводу певна частина реалізується і не потребує транспортування. За таких умов математична модель задачі можна представити наступним чином:

Знайти оптимальні стратегії гравців  $X_{i_0}$  і  $Y_{j_0}$ , де

- $X_{i_0}$  знаходиться з виразу  $F(X_{i_0}, Y_{j_1}) = \max_i \min_j F(X_i, Y_j)$ ;
- $Y_{j_0}$  – з виразу  $F(X_{i_1}, Y_{j_0}) = \min_j \max_i F(X_i, Y_j)$ ;
- $a_{ij} (\forall i \in J_m \quad \forall j \in J_n)$  – задані числа;

при обмеженнях:

$$1. \quad \bar{X} = (x_1, x_2, \dots, x_{m-1}) \in E_M^{m-1}(P^x); \quad X = (x_1, \dots, x_m): \sum_{i=1}^m x_i < 1;$$

вектор  $P^x = (P_1^x, P_2^x, \dots, P_M^x)$  задовольняє умовам  $P_i^x \geq 0 \quad \forall i \in J_M$ ;

$$2. \quad Y = (y_1, y_2, \dots, y_n) \text{ задовольняє умовам } \forall j \in J_n \quad y_j \geq 0; \quad \sum_{j=1}^n y_j = 1.$$

Задача, що розглянута вище є задачею ігрового типу з комбінаторними обмеженнями на множині розміщень, що накладаються на стратегії першого гравця гравця. Основною відмінністю від задачі 2 є те що сума вектора

переставлень менше одиниці, тому що продукція яка виробляється на підприємстві не розвозиться повністю по магазинам, а частина реалізується безпосередньо на підприємстві.

#### Задача 4.

Розглянемо ще один випадок задачі 2, припустимо, що перше підприємство має свої невеликі хлібопекарні у цьому ж місті і кожного ранку розвозить і свою продукцію також і продукцію цих підприємств. При такій постановці задачі математична модель задачі матиме вигляд:

Знайти оптимальні стратегії гравців  $X_{i_0}$  і  $Y_{j_0}$ , де

- $X_{i_0}$  знаходиться з виразу  $F(X_{i_0}, Y_{j_1}) = \max_i \min_j F(X_i, Y_j)$ ;
- $Y_{j_0}$  – з виразу  $F(X_{i_1}, Y_{j_0}) = \min_j \max_i F(X_i, Y_j)$ ;
- $a_{ij}$  ( $\forall i \in J_m \quad \forall j \in J_n$ ) – задані числа;

при обмеженнях:

$$1. \quad \bar{X} = (x_1, x_2, \dots, x_{m-1}) \in E_M^{m-1}(P^x); \quad X = (x_1, \dots, x_m): \sum_{i=1}^m x_i > 1;$$

вектор  $P^x = (P_1^x, P_2^x, \dots, P_M^x)$  задовольняє умовам  $P_i^x \geq 0 \quad \forall i \in J_M$ ;

$$2. \quad Y = (y_1, y_2, \dots, y_n) \text{ задовольняє умовам } \forall j \in J_n \quad y_j \geq 0; \quad \sum_{j=1}^n y_j = 1.$$

Дана математична модель описує ігрову задачу в якій на стратегію першого гравця накладаються комбінаторні обмеження, що визначаються множиною розміщень, причому сума вектора множини більше 1.

Для розв'язання задач 3 та 4 сформульовано алгоритм (див. пункт 2.2), який було програмно реалізовано (див. пункт 3.2).

### 1.5 Методи розв'язування задач комбінаторної оптимізації

Клас задач, що розглядається є новим і поєднує в собі риси задач комбінаторної оптимізації і теорії ігор. Для розв'язування таких задач потрібно модифікувати вже існуючі методи, щоб дізнатися, які з них можна використовувати для розв'язування саме такого класу задач [9-11].

Звичайно одним із методів розв'язування таких задач є метод повного перебору, але коли йдеться про задачі великої вимірності, то він звичайно є дуже громістким і займає багато часу, отже, необхідно розробляти нові методи, які б давали розв'язок з меншими затратами часу.

Найбільшого поширення набули так звані комбінаторні методи, які об'єднують методи гілок та меж, метод послідовного аналізу варіантів, методи побудови послідовності розв'язків, методи локальної оптимізації, метод динамічного програмування, метод послідовних розрахунків, апроксимаційно-комбінаторний та інші.

Основна ідея комбінаторних методів полягає у переході від повного перебору скінченної множини розв'язків до скороченого, напрямленого.

Одним із підходів у розв'язанні комбінаторних задач є евристичний метод. Евристичні алгоритми, хоч і дозволяють врахувати будь-які обмеження, а також звичайно швидко дають розв'язок, але не гарантують його оптимальності. Відхилення від глобального екстремуму оцінити, як правило, важко, а часом неможливо.

Одним із точних та розповсюджених методів в евклідовій комбінаторній оптимізації є метод відсікань [5]. Розроблений метод комбінаторного відсікання для задач оптимізації на евклідових комбінаторних множинах, що збігаються з множиною вершин своєї опуклої оболонки (це дозволяє уникнути труднощів застосування методів відсікання в цілочисловій оптимізації, що пов'язані з наявністю внутрішніх цілочислових точок допустимої області). В цьому методі відсікання проводиться через вершини допустимої області допоміжної задачі, суміжними з вершиною, яка відсікається. При цьому відсікається максимально можлива частина многогранника, і всі елементи евклідової комбінаторної множини є вершинами одержаного після відсікання многогранника. Для задач комбінаторної оптимізації на переставленнях метод комбінаторного відсікання розроблений як для повністю комбінаторних задач, так і для частково комбінаторних (в яких не всі змінні, від яких залежить цільова функція, належать комбінаторній множині).

## 1.6 Метод знаходження оптимальної стратегії гравця в матричних задачах теорії ігор з комбінаторними обмеженнями, які визначаються переставленнями

Нехай потрібно знайти оптимальну стратегію гравця  $X_{i_0}$  :  
 $F(X_{i_0}, Y_{j_1}) = \max_i \min_j F(X, Y)$ , в матричних задачах теорії ігор з комбінаторними обмеженнями, що визначаються переставленнями ( $X \in E_m(P^x)$ ;  $Y \in R^n$ ), зведеться до наступної задачі комбінаторної оптимізації на переставленнях:

$$\begin{aligned} \sum_{i=1}^m a_{i1} x_i - x_{m+1} &\rightarrow \max \\ \sum_{i=1}^m (a_{i1} - a_{ij}) x_i - x_{m+1} + x_{m+j} &= 0 \quad \forall j \in J_n^2; \\ x_{m+j} &\geq 0 \quad \forall j \in J_n; \\ X = (x_1, x_2, \dots, x_m) &\in E_m(P^x); \quad \forall i \in J_m \quad P_i^x \geq 0; \quad \sum_{i=1}^m P_i^x = 1, \end{aligned}$$

де  $E_m(P^x)$  – множина переставлень з елементів вектора  $P^x$ ;

або, з огляду на те, що  $x_{m+j} \geq 0 \quad \forall j \in J_n$  і позначивши  
 $c_i = a_{i1}$ ,  $b_{i(j-1)} = (a_{i1} - a_{ij}) \quad \forall j \in J_n^2$ :

$$\begin{aligned} F &= \sum_{i=1}^m c_i x_i - x_{m+1} \rightarrow \max \\ \sum_{i=1}^m b_{ij} x_i &\leq x_{m+1} \quad \forall j \in J_{n-1}; \\ x_{m+1} &\geq 0 \end{aligned} \tag{1.15}$$

$$X = (x_1, x_2, \dots, x_m) \in E_m(P^x); \quad \forall i \in J_m \quad P_i^x \geq 0; \quad \sum_{i=1}^m P_i^x = 1.$$

Присвоїмо для зручності посилання нерівностям номер:

$$\sum_{i=1}^m b_{ij} x_i \leq x_{m+1} \quad \forall j \in J_{n-1}; \tag{1.16}$$

Розглянемо метод знаходження оптимальної стратегії гравця в матричних задачах теорії ігор з комбінаторними обмеженнями, які визначені переставленнями:



1. Обчислимо  $d_1 = \max_{\substack{j \in J_{n-1} \\ x \in E_m(P^x)}} \left\{ \sum_{i=1}^m b_{ij} x_i \right\}$ , (значення  $x_{m+1}$ , при якому всі точки

переставного многогранника будуть задовольняти нерівностям

$\sum_{i=1}^m b_{ij} x_i \leq x_{m+1} \quad \forall j \in J_{n-1}$ ; і  $d^1 = \min_{\substack{j \in J_{n-1} \\ x \in E_m(P^x)}} \left\{ \sum_{i=1}^m b_{ij} x_i \right\} - \varepsilon, \varepsilon > 0$  (значення  $x_{m+1}$ , при якому

жодна точка переставного многогранника не буде задовольняти нерівностям).

Впорядкуємо коефіцієнти  $b_{ij}$  при фіксованих  $j$  і елементи множини  $P_i^x$  по

зростанню: отримані елементи позначимо  $\beta_{ij}$  і  $\alpha_i$  відповідно. Тоді відповідно

$$d_1 = \max_{\substack{j \in J_{n-1} \\ x \in E_m(P^x)}} \left\{ \sum_{i=1}^m b_{ij} x_i \right\} = \sum_{i=1}^m \beta_{ij} \alpha_i \quad \text{і} \quad d^1 = \min_{\substack{j \in J_{n-1} \\ x \in E_m(P^x)}} \left\{ \sum_{i=1}^m b_{ij} x_i \right\} - \varepsilon = \sum_{i=1}^m \beta_{ij} \alpha_{m-i+1} - \varepsilon.$$

2. Спочатку вибираємо  $x_{m+1} = d_1$ , тоді всі точки  $X = (x_1, x_2, \dots, x_m) \in E_m(P^x)$

будуть задовольняти нерівностям  $\sum_{i=1}^m b_{ij} x_i \leq x_{m+1} \quad \forall j \in J_{n-1}$ .

3. Розв'язуємо лінійну, безумовну, повністю комбінаторну задачу комбінаторної оптимізації на переставленнях:

$$f = \sum_{i=1}^m c_i x_i \rightarrow \max$$

$$X = (x_1, x_2, \dots, x_m) \in E_m(P^x).$$

Для цього впорядкуємо коефіцієнти  $c_i$  по зростанню, отримані елементи

позначимо  $\gamma_i$ . Тоді отримаємо  $f_1^* = \max_{x \in E_m(P^x)} f(X) = \sum_{i=1}^m \gamma_i \alpha_i$ . Нехай

$X_1^* = (x_1, x_2, \dots, x_m) \in E_m(P^x)$  — точка переставного многогранника, в якій досягається це максимальне значення функції  $f(X)$ .

4. Для точки  $X_1^* = (x_1, x_2, \dots, x_m)$  знаходимо значення  $d_2 = x_{m+1} = \max_{1 \leq j \leq n-1} \left\{ \sum_{i=1}^m b_{ij} x_i \right\}$  і

значення цільової функції вихідної задачі  $F_1 = f_1^* - d_2$  (цільова функція вихідної

задачі має вигляд  $F = \sum_{i=1}^m c_i x_i - x_{m+1} = f - x_{m+1}$ ). Оскільки в точці  $X_1^*$  досягається

максимальне (по  $X \in E_m(P^x)$ ) значення функції  $f = \sum_{i=1}^m c_i x_i$ , то в усіх точках

$X = (x_1, x_2, \dots, x_m) \in E_m(P^x)$ , для яких  $x_{m+1} \geq d_2$ , значення цільової функції  $F$  не буде перевищувати  $F_1$  (дійсно, оскільки  $F = f - x_{m+1}$ ,  $f_1^* = \max_{x \in E_m(P^x)} f(X)$  для тих точок, для яких  $x_{m+1} \geq d_2$ , або  $-x_{m+1} \leq -d_2$ , тому для них  $F = f - x_{m+1} \leq f_1^* - d_2 = F_1$ ). Тому далі розглядаємо точки  $X = (x_1, x_2, \dots, x_m) \in E_m(P^x)$ , для яких  $x_{m+1} < d_2$ , тобто величині  $d_2$  присвоюємо значення  $d_2 - \varepsilon$  і розглядаємо  $x_{m+1} \leq d_2$ .

5. Нехай  $s = 2$ .

6. Перевіряємо умову  $d^1 \geq d_s$ .  $d^1$  – таке значення  $x_{m+1}$ , при якому жодна точка переставного многогранника не буде задовольняти нерівностям

$$\sum_{j=1}^i x_{\alpha_j} \leq \sum_{j=1}^i g_j,$$

$$\alpha_j \in J_{k-1}, \alpha_j \neq \alpha_r \quad \forall j \neq r, j, r \in J_i \quad \forall i \in J_{k-1},$$

$$l_h(x) \leq 0 \quad \forall h \in J_t.$$

Якщо ця умова виконується, то жодна точка  $X = (x_1, x_2, \dots, x_m) \in E_m(P^x)$  не задовольняє нерівностям, переходимо на крок 9, якщо ні – на крок 7.

7. Розв'язуємо лінійну, умовну, повністю комбінаторну задачу комбінаторної оптимізації на переставленнях:

$$f = \sum_{i=1}^m c_i x_i \rightarrow \max$$

$$\sum_{i=1}^m b_{ij} x_i \leq d_s; \quad (1.17)$$

$$X = (x_1, x_2, \dots, x_m) \in E_m(P^x);$$

Для цього можна використовувати метод комбінаторного відсікання для повністю комбінаторних задач. Якщо розв'язок знайшли, то переходимо на крок 8.

8. Нехай  $f_s^*$ ,  $X_s^*$  – оптимальне значення цільової функції. Якщо ж розв'язка не отримали, зменшуємо  $s$  на 1, переходимо на крок 9.

8. Для точки  $X_s^*$  знаходимо значення  $d_{s+1} = x_{m+1} = \max_{\substack{1 \leq j \leq n-1 \\ x_j \in X_s^*, i \in J_m}} \left\{ \sum_{i=1}^m b_{ij} x_i \right\}$  і значення

цільової функції вихідної задачі  $F_s = f_s^* - d_{s+1}$ . Присвоюємо величині  $d_{s+1}$  значення  $d_{s+1} - \varepsilon$ . Збільшуємо  $s$  на 1, переходимо на крок 6.

9. Вибираємо  $F^* = \max_{1 \leq i \leq s-1} \{F_i\}$ ,  $X^* = \arg \max_{1 \leq i \leq s-1} \{F_i\}$  — розв’язок вихідної задачі (оптимальна стратегія першого гравця).

В подальших розділах розглядається поширення даного метода на задачі ігрового типу з комбінаторними обмеженнями, що визначаються розміщеннями, на стратегії одного гравця.

## 2 ТЕОРЕТИЧНА ЧАСТИНА

### 2.1 Ітераційний метод для знаходження оптимальної стратегії гравця, на стратегії якого накладаються комбінаторні обмеження, що визначаються розміщеннями

У [10] розглянуто ітераційний метод знаходження оптимальної стратегії першого гравця з обмеженнями, що визначаються переставленнями. Даний метод ідейно близький методу Брауна-Робінсона для розв'язування класичних матричних ігор. Розповсюдимо даний метод на ігрові задачі в яких на стратегії одного гравця накладаються комбінаторні обмеження визначені переставленнями. Всі кроки даного методу спираються на припущення, що гравці прагнуть збільшити свій виграш (зменшити програш) та роблять ходи за принципом "майбутнє схоже на минуле", враховують всі зроблені ходи. Елемент  $a_{ij}$  матриці  $A$  – це різниця прибутків першого і другого підприємств, в тому випадку, коли перше підприємство завозить продукцію у свій фірмовий магазини за номером  $i$ , а друге підприємство – за номером  $j$ . Тоді перший гравець (перше підприємство) має за мету зменшити свій програш, а другий гравець – збільшити свій виграш.

Всі розрахунки (ітерації) будемо заносити в таблицю (вигляду Додатку А), в якій  $N$  – номер кроку,  $X$  – вибрана стратегія першого гравця,  $B_1, B_2, \dots, B_n$  – стратегії другого гравця,  $B_1X, B_2X, \dots, B_nX$  – скалярний добуток векторів,  $N\bar{v}$  – максимальний накопичений виграш (максимальний з накопичених скалярних добутоків  $B_1X, \dots, B_nX$ , які зберігаються в рядку  $SUM$ ),

$$\bar{v} = \frac{N\bar{v}}{N}, \quad (2.1)$$

$j$  – вибрана стратегія другого гравця,  $A_1, A_2, \dots, A_m$  – програш першого гравця для його чистої стратегії  $1, 2, 3, \dots, m$  відповідно при вибраній стратегії другого; числа в рядку з назвою  $SUM$  – накопичені суми значень з попереднього рядка з

назвою  $sum$ ,  $NextX$  – вибрана стратегія першого гравця,  $N_v$  – мінімальний накопичений програш (скалярний добуток елементів векторів  $SUM$  і  $NextX$ ),

$$\underline{v} = \frac{N_v}{N}, \quad \bar{v} = \frac{N_{\bar{v}}}{N}, \quad (2.2)$$

$$v^* = \frac{\bar{v} + \underline{v}}{2}. \quad (2.3)$$

На першому кроці ітераційного методу перший гравець вибирає навмання свою першу стратегію. Другий гравець аналізує виграш, який він отримає при кожній своїй стратегії, та вибирає стратегію з метою максимізувати виграш (починаючи з другої ітерації накопичений виграш) – значення в рядку  $SUM$ . Перший гравець вибирає свою наступну стратегію  $NextX$ , яка б мінімізувала програш (починаючи з другої ітерації накопичений програш) – скалярний добуток елементів векторів  $SUM$  і  $NextX$ . Цей процес продовжується далі.

Якщо перший гравець зробив  $S$  ходів, то другий гравець вибирає свою стратегію  $j_s$  з метою максимізувати накопичений виграш – максимальне зі значень відповідного рядка  $SUM$ . Перший гравець на кроці  $S+1$  вибирає свою стратегію  $Next X_{s+1}$  таким чином, щоб мінімізувати накопичений програш – скалярний добуток елементів вектора  $Next X_{s+1}$  і відповідного рядка  $SUM$ .

Розглянемо роботу ітераційного метода на прикладі.

Приклад.

$$\text{Нехай } M=10, m=3, n=2, P^x = \{0.1, 0.1, 0.3, 0.4, 0.5, 0.6, 0.7, 0.7, 0.9, 0.9\}, A = \begin{pmatrix} 7 & 3 \\ 1 & 5 \\ 3 & 4 \end{pmatrix}$$

Розв'язати задачу 2 з пункту 1.5 з цими даними, нагадаємо, що накладається умова:  $X = (x_1, \dots, x_m): \sum_{i=1}^m x_i = 1$ . За даної умови спочатку потрібно сформулювати розміщення  $X$  з множини  $P^x$ , причому сума елементів кожного з них повинна дорівнювати 1.

$$N_{10}^3(P^x) = \{(0,4; 0,1; 0,5), (0,4; 0,5; 0,1), (0,1; 0,4; 0,5), (0,1; 0,5; 0,4), (0,5; 0,1; 0,4), (0,5; 0,4; 0,1),$$

$(0,4; 0,5; 0,1), (0,4; 0,1; 0,5), (0,5; 0,4; 0,1), (0,5; 0,1; 0,4), (0,1; 0,5; 0,4), (0,1; 0,4; 0,5), (0,1; 0,3; 0,6), (0,1; 0,6; 0,3), (0,3; 0,1; 0,6), (0,3; 0,6; 0,1), (0,6; 0,3; 0,1), (0,6; 0,1; 0,3), (0,3; 0,6; 0,1), (0,3; 0,1; 0,6), (0,6; 0,3; 0,1), (0,6; 0,1; 0,3), (0,1; 0,6; 0,3), (0,1; 0,3; 0,6)\}$ .

Складемо таблицю (Додаток А), в яку будемо заносити результати розрахунків на кожній ітерації метода. В якості першої стратегії першого гравця виберемо стратегію навмання – наприклад,  $(0,1; 0,3; 0,6)$ .

Зробимо 40 ітерацій. Після 40-ї ітерації методу виявилось, що стратегія  $(0,1, 0,3, 0,6)$  першого гравця використовувалась 15 разів, стратегія  $(0,6, 0,1, 0,3)$  – 10 разів,  $(0,5, 0,1, 0,4)$  - 2 рази,  $(0,3, 0,1, 0,6)$  - 12 разів,  $(0,1, 0,6, 0,3)$  - 1 раз. Оптимальною стратегією першого гравця буде стратегія з максимальною ймовірністю 0.375, тобто стратегія  $(0,1, 0,3, 0,6)$ .

Для другого гравця: перша стратегія використовувалась 10 разів, а друга – 30 разів, ймовірність застосування першої стратегії становить 0,25, а другої – 0,75. Мішаною стратегією другого гравця буде стратегія  $(0,25; 0,75)$ .

## 2.2 Алгоритм методу

Як було зазначено, розрахунки робляться в припущенні, що гравці прагнуть збільшити свій виграш (зменшити програш). Вважається, що вони не знають своїх оптимальних стратегій. Ходи гравці роблять за принципом: майбутнє схоже на минуле, враховуючи всі зроблені ходи.

Розглянемо послідовність кроків алгоритму даного методу:

1. Першу стратегію  $i$  першого гравця обираємо випадковим чином (в створеній програмі - першу).
2. Обчислюємо скалярні добутки векторів стратегій другого гравця на вектор обраної стратегії першого гравця, та відповідні суми  $sum$  скалярних добутків.
3. Обраховуємо накопичені суми  $SUM$  скалярних добутків.

4. Обираємо стратегію другого гравця за критерієм отримання максимального виграшу (стратегію з максимальною накопиченою сумою).
5. Стратегія  $NextX$  другого гравця обирається з умови отримання мінімального програшу [1].
6. Обчислюємо  $N_{\underline{v}}$  – мінімальний накоплений програш, як скалярний добуток елементів векторів накопиченої суми  $SUM$  програвів другого гравця і обраної стратегії  $NextX$  другого гравця.
7. За формулами (2.1)-(2.3) обчислюємо  $\underline{v}$ ,  $\bar{v}$  та  $v^*$ .
8. Перевіряємо критерій завершення роботи алгоритму (рівність максимального накопленого виграшу першого гравця мінімальному накопленому програшу другого гравця) і, якщо він виконується, роботу алгоритму завершено, інакше переходимо на пункт 2 алгоритму, обравши за стратегію першого гравця стратегію  $NextX$ . Робота алгоритму також може бути закінчена, якщо проведена вказана кількість ітерацій.

### 2.3 Аналіз часу роботи алгоритму

В ході виконання роботи було проведено чисельний експеримент з метою виявлення залежності часу обчислень від вимірності задачі. Дослідження проводилися на великій кількості тестових прикладів за допомогою вбудованого генератора початкових даних. Результати досліджень для задач (1) – (4) з пункту 1.5 наведені в таблицях 2.1 – 2.5 відповідно, для кожної вимірності було розв’язано 100 задач, вимірності  $m \times n$ , причому  $m = n$ , з максимальною кількістю ітерації 100000, та з вектором множини з якої формувались розміщення –  $M$ . На рисунках 3.1-3.5 наведено діаграми залежності часу роботи програми від вимірності задачі.

Таблиця 2.1. Залежність часу обчислень від вимірності задачі (на переставленнях)

| $m = n$ | Проведено ітерацій | Заг.час (мс) | Сер.час задачі (мс) | Сер. ітерації (мс) |
|---------|--------------------|--------------|---------------------|--------------------|
| 10      | 10000              | 1725         | 17                  | 0,172              |
| 20      | 9813               | 1755         | 18                  | 0,179              |
| 30      | 10000              | 1779         | 18                  | 0,178              |
| 50      | 9804               | 1735         | 17                  | 0,177              |
| 75      | 10000              | 1839         | 18                  | 0,184              |
| 100     | 9907               | 1753         | 18                  | 0,177              |
| 150     | 100000             | 1761         | 18                  | 0,018              |
| 200     | 9710               | 1803         | 18                  | 0,186              |
| 300     | 9617               | 1849         | 18                  | 0,192              |
| 500     | 9901               | 2006         | 20                  | 0,203              |
| 625     | 9514               | 2192         | 22                  | 0,230              |
| 750     | 9411               | 2312         | 23                  | 0,246              |
| 875     | 9735               | 2491         | 25                  | 0,256              |
| 1000    | 9756               | 2815         | 28                  | 0,289              |



Рисунок 2.1 - Залежність часу обчислень від вимірності задачі (на переставленнях)



Таблиця 2.2. Залежність часу обчислень від вимірності задачі (на розміщеннях при  $\sum_{i=1}^m x_i = 1$ )

| $m = n = M$ | Проведено ітерацій | Заг. час (мс) | Сер. час задачі (мс) | Сер. час ітерації (мс) |
|-------------|--------------------|---------------|----------------------|------------------------|
| 10          | 9901               | 3064          | 30                   | 0,310                  |
| 20          | 9902               | 4174          | 41                   | 0,422                  |
| 30          | 9901               | 3458          | 34                   | 0,349                  |
| 50          | 10000              | 4032          | 40                   | 0,403                  |
| 75          | 10000              | 5118          | 51                   | 0,512                  |
| 100         | 9921               | 6639          | 66                   | 0,669                  |
| 150         | 10000              | 11139         | 111                  | 1,114                  |
| 200         | 10000              | 17311         | 173                  | 1,731                  |
| 300         | 10000              | 33845         | 338                  | 3,385                  |
| 500         | 10000              | 88538         | 885                  | 8,854                  |
| 625         | 10000              | 135711        | 1357                 | 13,571                 |
| 750         | 10000              | 193404        | 1934                 | 19,340                 |
| 875         | 10000              | 263898        | 2638                 | 26,390                 |
| 1000        | 10000              | 337232        | 3372                 | 33,723                 |

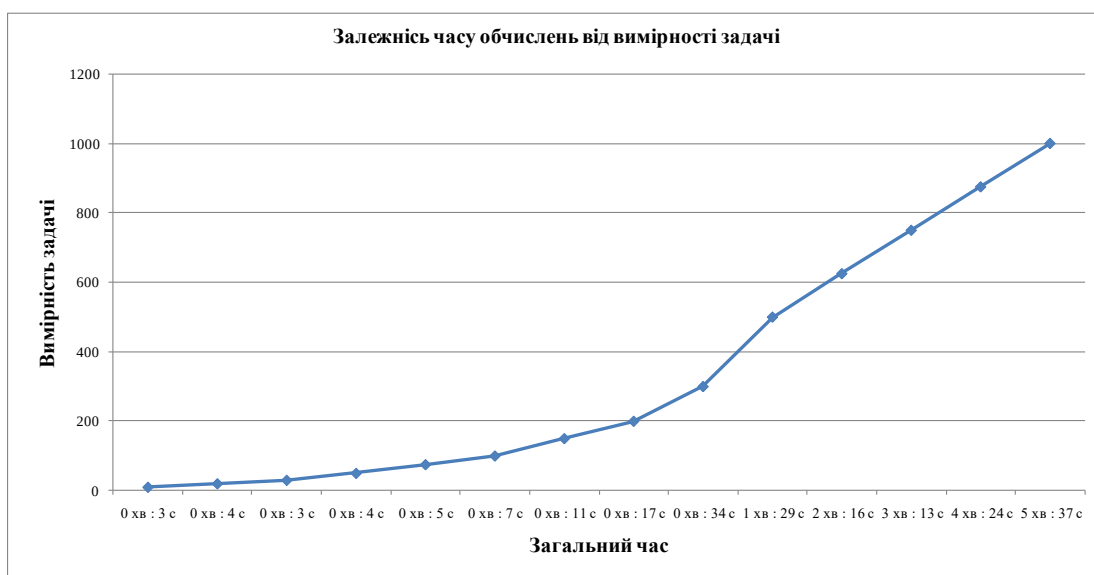


Рисунок 2.2 - Залежність часу обчислень від вимірності задачі (на

розміщеннях при  $\sum_{i=1}^m x_i = 1$ )

Таблиця 2.3. Залежність часу обчислень від вимірності задачі (на розміщеннях при  $\sum_{i=1}^m x_i > 1$ )

| $m = n = M$ | Проведено ітерацій | Заг. час (мс) | Сер. час задачі (мс) | Сер. час ітерації (мс) |
|-------------|--------------------|---------------|----------------------|------------------------|
| 10          | 9366               | 4039          | 40                   | 0,431                  |
| 20          | 9466               | 3144          | 31                   | 0,332                  |
| 30          | 9448               | 3355          | 34                   | 0,355                  |
| 50          | 9696               | 4029          | 40                   | 0,416                  |
| 75          | 9714               | 5095          | 51                   | 0,524                  |
| 100         | 9814               | 6654          | 67                   | 0,678                  |
| 150         | 9748               | 11040         | 110                  | 1,133                  |
| 200         | 9755               | 16990         | 170                  | 1,742                  |
| 300         | 9710               | 32940         | 329                  | 3,392                  |
| 500         | 9847               | 87244         | 872                  | 8,860                  |
| 625         | 9761               | 132566        | 1326                 | 13,581                 |
| 750         | 9692               | 187727        | 1877                 | 19,369                 |
| 875         | 9973               | 263327        | 2633                 | 26,404                 |
| 1000        | 9811               | 331065        | 3311                 | 33,744                 |

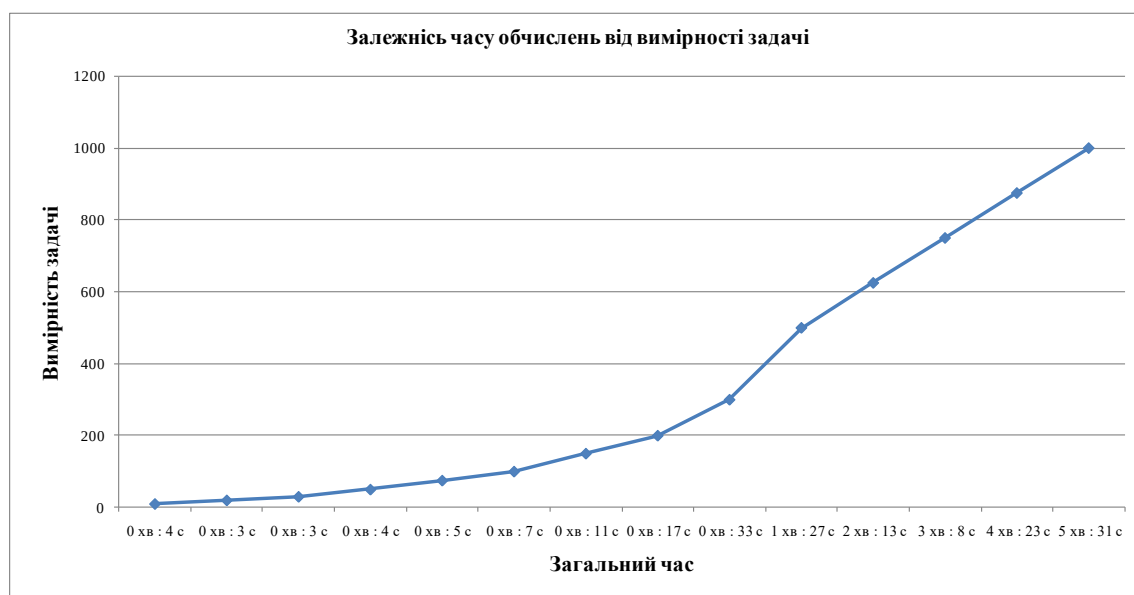


Рисунок 2.3 - Залежність часу обчислень від вимірності задачі (на розміщеннях при  $\sum_{i=1}^m x_i > 1$ )

Таблиця 2.4. Залежність часу обчислень від вимірності задачі (на розміщеннях при  $\sum_{i=1}^m x_i < 1$ )

| $m = n = M$ | Проведено ітерацій | Заг. час (мс) | Сер. час задачі (мс) | Сер. час ітерації (мс) |
|-------------|--------------------|---------------|----------------------|------------------------|
| 10          | 10000              | 3110          | 31                   | 0,311                  |
| 20          | 10000              | 3214          | 32                   | 0,321                  |
| 30          | 10000              | 3439          | 34                   | 0,344                  |
| 50          | 9901               | 3996          | 40                   | 0,404                  |
| 75          | 9901               | 5080          | 51                   | 0,513                  |
| 100         | 10000              | 6633          | 66                   | 0,663                  |
| 150         | 10000              | 11099         | 111                  | 1,110                  |
| 200         | 10000              | 17304         | 173                  | 1,730                  |
| 300         | 10000              | 33758         | 338                  | 3,376                  |
| 500         | 10000              | 88253         | 883                  | 8,825                  |
| 625         | 10000              | 135470        | 1355                 | 13,547                 |
| 750         | 10000              | 192932        | 1929                 | 19,293                 |
| 875         | 10000              | 263709        | 2637                 | 26,371                 |
| 1000        | 10000              | 336633        | 336633               | 33,663                 |

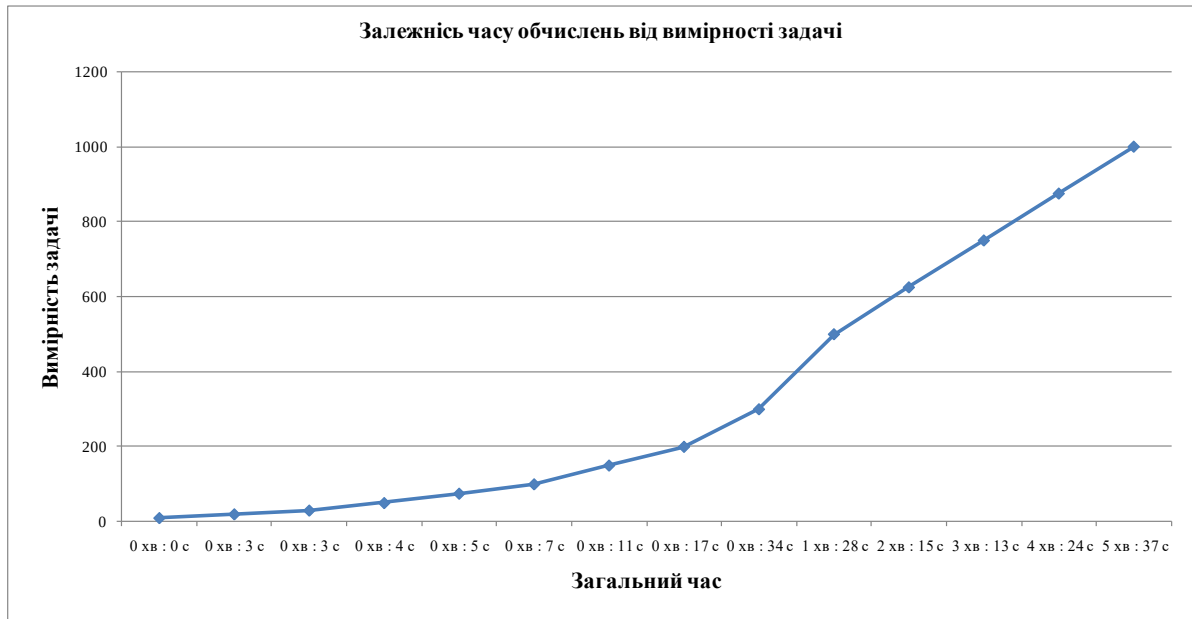


Рисунок 2.4 - Залежність часу обчислень від вимірності задачі (на розміщеннях

$$\text{при } \sum_{i=1}^m x_i < 1)$$

Починаючи з вимірності 200 спостерігається поступове збільшення часу обчислення. При  $\sum_{i=1}^m x_i > 1$  час в середньому більший через більшу кількість генерованих розміщень, що відповідають заданій умові.

## РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА

### 3.1 Блок-схема алгоритму

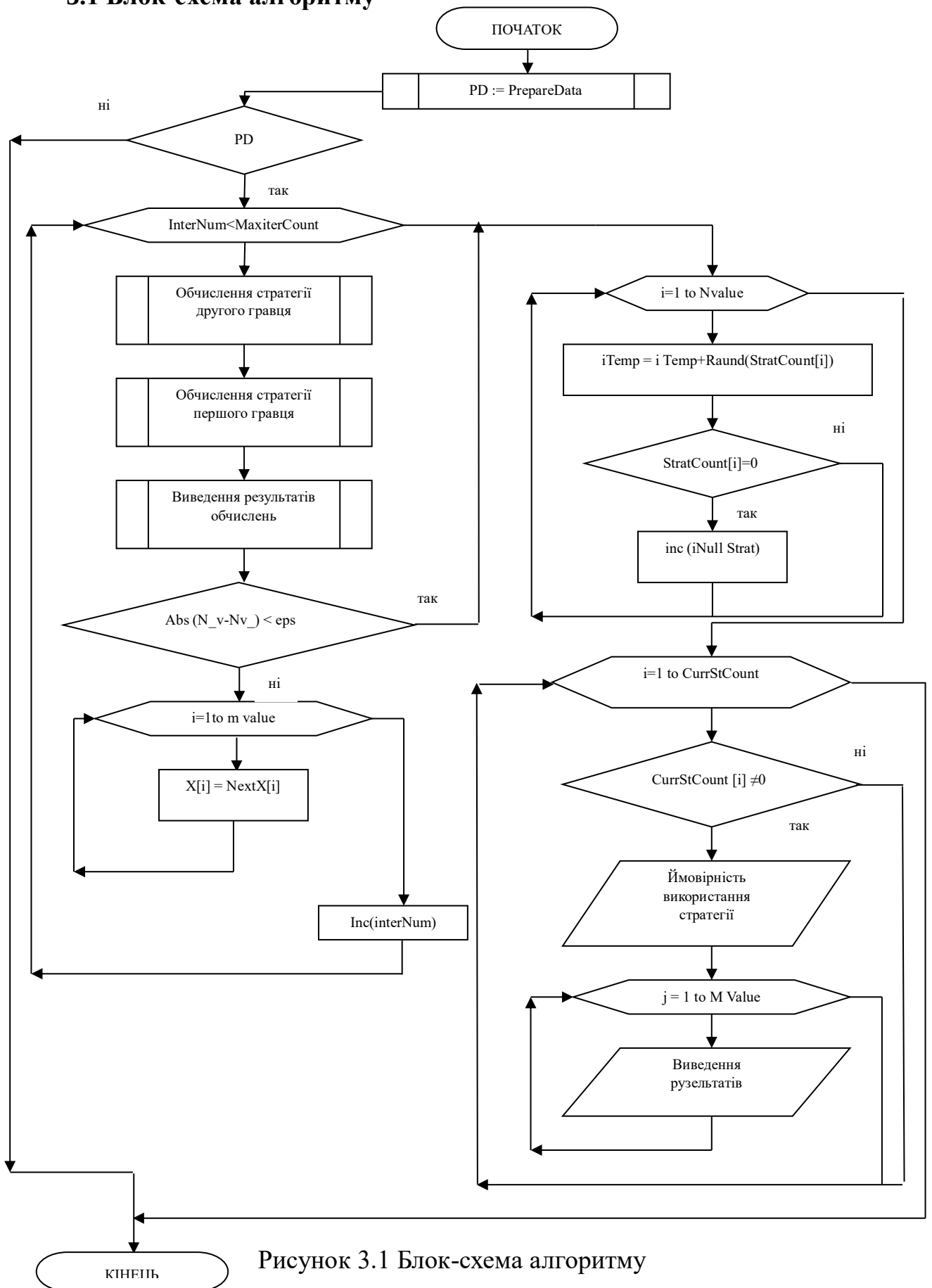


Рисунок 3.1 Блок-схема алгоритму

Опис всіх змінних знаходиться в пункті 3.2.

Блок-схема функції PrepareRozm, генерація та підготовка розміщень для розв'язування задачі (рис. 3.2). Опис всіх змінних знаходиться в пункті 3.2.

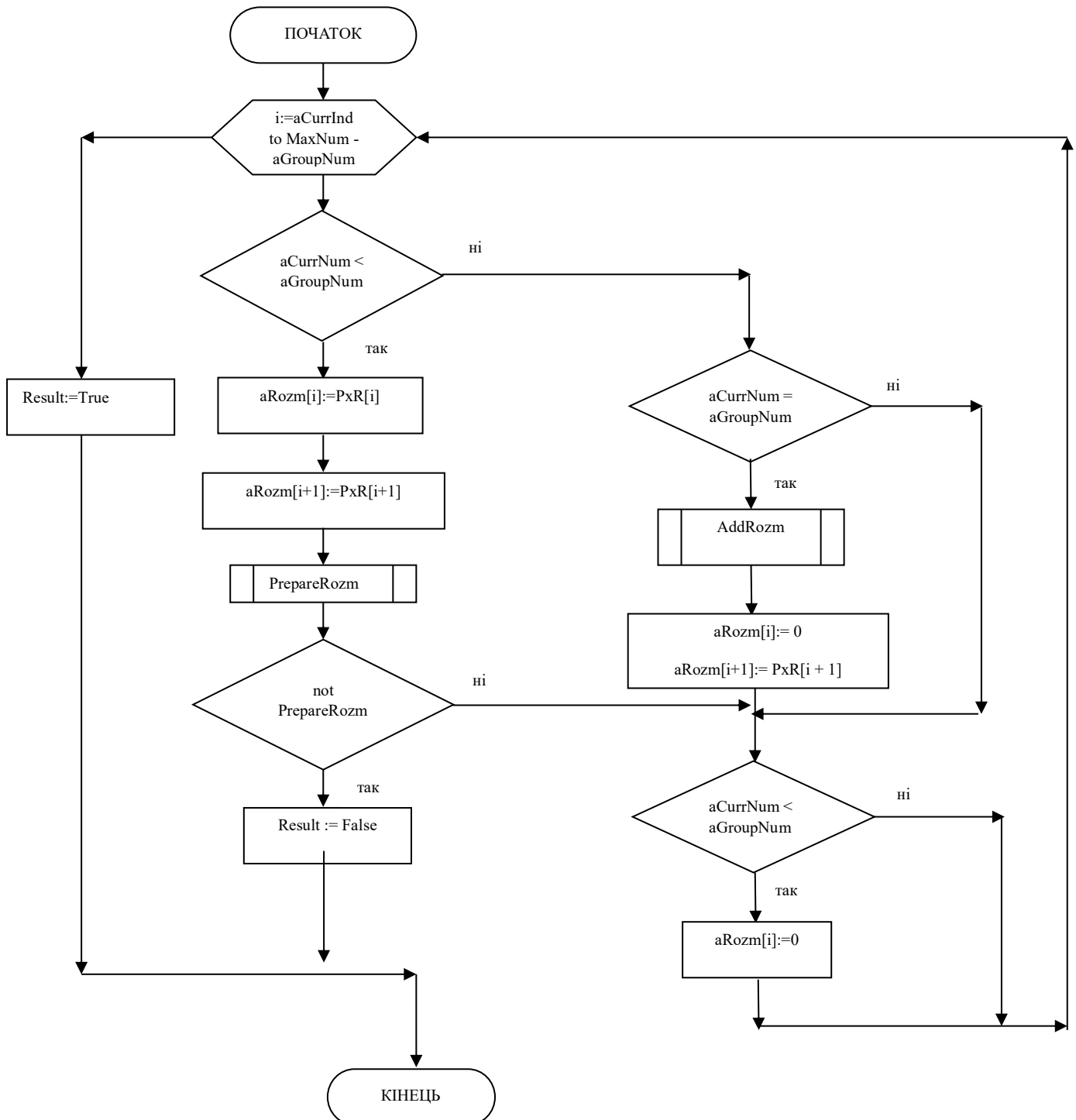


Рисунок 3.2 Блок-схема процедуры *PrepareRozm*

Процедура *GenerPx*, в якій генерується вектор переставлень на стратегії першого гравця (рис. 3.3). Опис всіх змінних знаходиться в пункті 3.2.

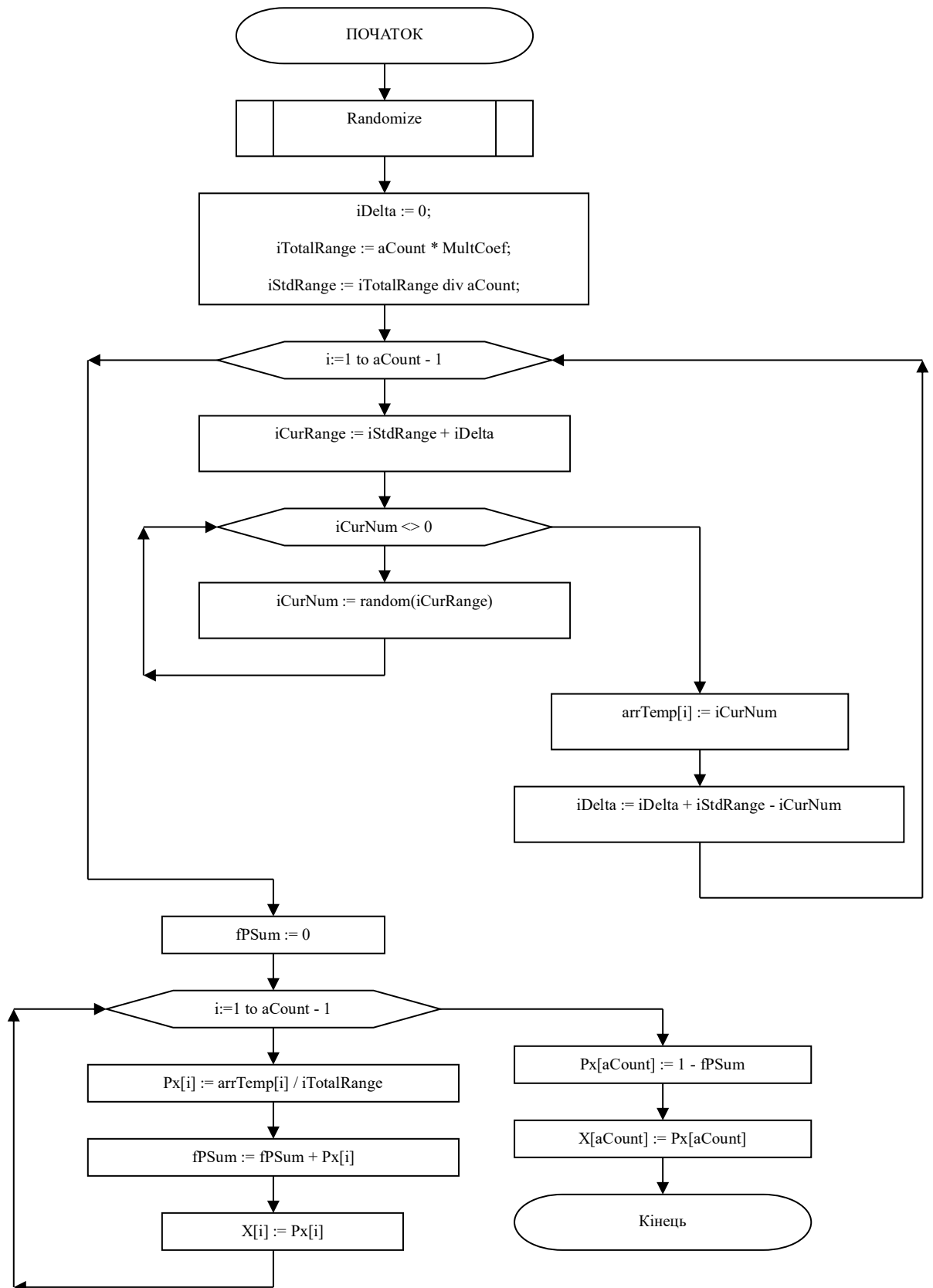


Рисунок 3.3 Блок-схема процедури *GenerPx*

Процедура Eval\_B\_Part, призначена для визначення параметрів другого гравця (рис. 3.4). Опис всіх змінних знаходиться в пункті 3.2.

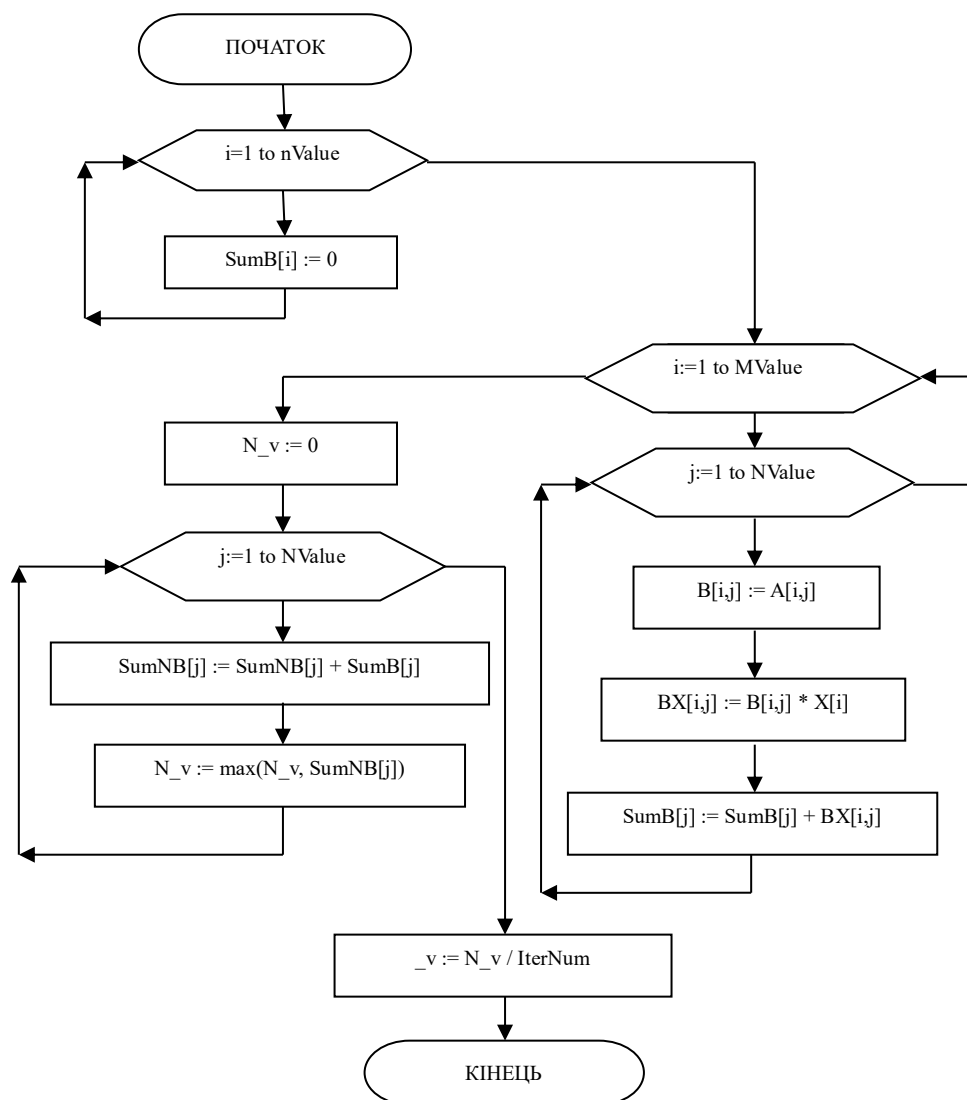


Рисунок 3.4 Блок-схема процедури *Eval\_B\_Part*



Процедура Eval\_A\_Part, обчислення параметрів і стратегій першого гравця (рис. 3.5). Опис всіх змінних знаходиться в пункті 3.2.

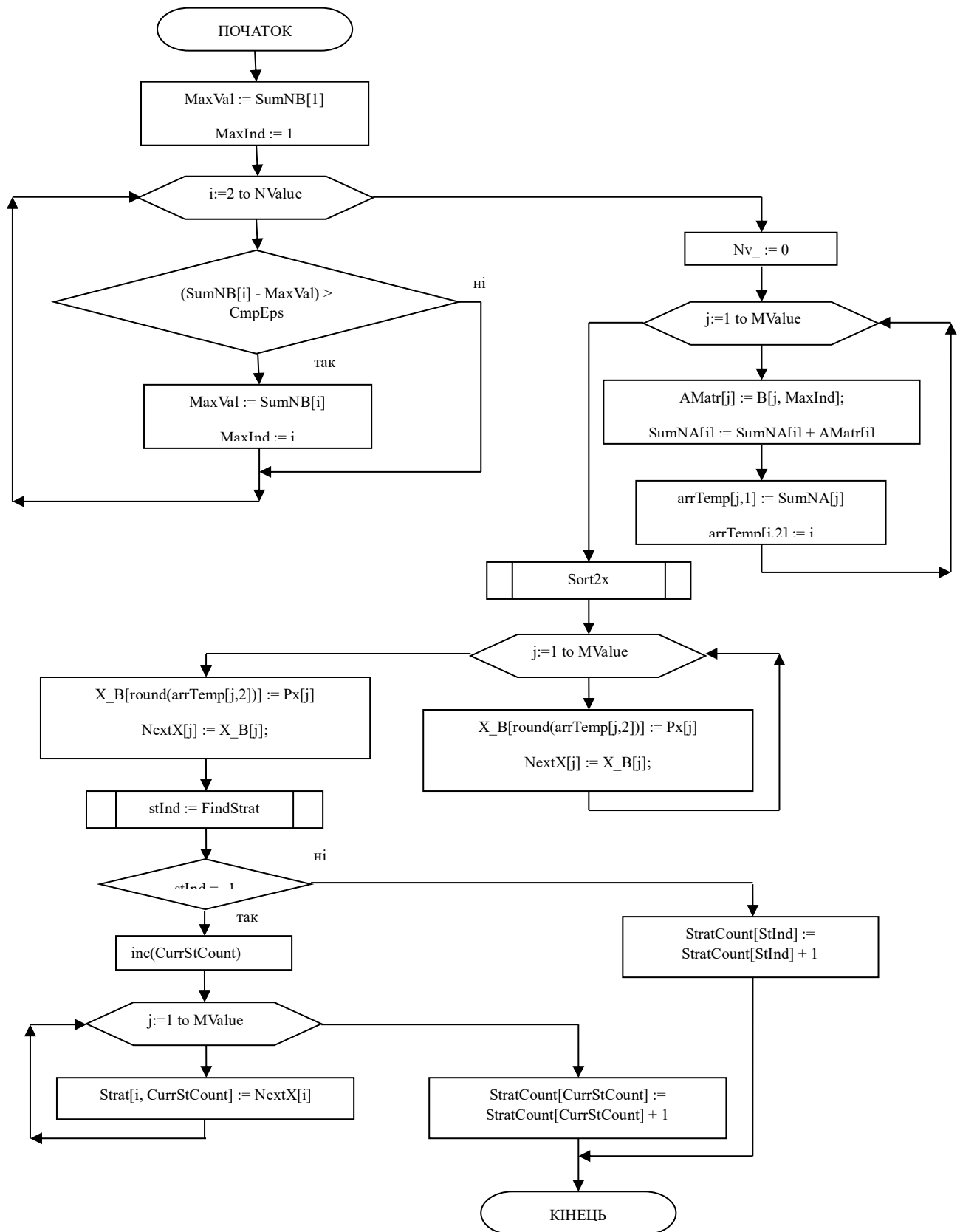


Рисунок 3.5 Блок-схема процедури Eval\_A\_Part

### 3.2 Опис програми

При написанні програми важливою вимогою була структурованість для можливості часткової модифікації та розширення функцій, поширення методів на інші класи комбінаторних множин. З огляду на це було створено велику кількість процедур та функцій:

- *GenerA* - функція, призначена для генерації початкової платіжної матриці. Матриця генерується випадковим чином (в залежності від налаштувань, за нормальним або рівномірним розподілом), зі значеннями в заданому користувачем інтервалі.
- *GenerPx* – процедура, в якій генерується вектор переставлень на стратегії першого гравця. Генеровані значення знаходяться на проміжку від 0 до 1, та в сумі складають 1.
- *GenerPxR* – процедура генерації вектора розміщень на стратегії першого гравця; генеровані значення знаходяться на проміжку від 0 до 1.
- *PrepareRozm* – рекурсивна функція побудови розміщень для задачі згідно обраних параметрів.
- *AddRozm* – процедура обробки кожного побудованого розміщення окремо.
- *FindStrat* – службова функція для пошуку використаних стратегій, якщо стратегія вже зустрічалася, то повертається її порядковий номер, інакше результат дорівнює -1.
- *Eval\_B\_Part* – процедура, призначена для визначення параметрів другого гравця.
- *Eval\_A\_Part* – процедура обчислення параметрів і стратегій першого гравця. При обчисленні спирається на результати, отримані в функції *Eval\_B\_Part*.
- *PrepareData* – функція, що використовується для читання вхідних даних з полів введення та присвоєння початкових значень всім

необхідним змінним. Результат «істина», якщо всі дані введено коректно та вдалося їх прочитати.

- *Sort1x* – процедура сортування типізованого одновимірного масиву за спаданням.
- *Sort2x* – процедура, призначена для сортування двовимірного масиву за зростанням (по першому стовпчику).
- *btCalcClick* – основна процедура обчислень, обробка події натискання клавіші початку обчислень.
- *CalcRozm* – процедура для проведення розв’язання задач на розміщеннях.
- *CalcPerest* – процедура проведення обчислень задач на переставленнях.
- *TasksAnalyze* – процедура, в якій реалізовано алгоритм аналізу залежності часу роботи програми від вимірності задач.
- *SaveData* – функція збереження початкових даних у файл. Результат «істина», якщо всі дані успішно збережені.
- *LoadData* – функція завантаження початкових даних з файлу.
- *PrintTableHeader* – процедура заповнення заголовку таблиці при виведенні результатів.
- *PrintTable* – процедура виведення вмісту таблиць на екран для кожної ітерації окремо.
- *ExportToExcel* – процедура, в якій реалізовано експорт результатів обчислень на робочий листок табличного процесора MS Excel;
- *ClearGrid* – процедура очищення таблиць виведення.
- *ExitFromProgram* – процедура обробки події виходу з програми.
- Модуль *HRTimer* містить в собі процедури та функції для реалізації інструменту високоточного таймеру (з точністю до  $10^{-8}$  мілісекунди).
- В модулі *HtmlUnit* розміщено функції та процедури, що застосовуються виведення розширених результатів обчислень у файл.

Текст програми наведено на диску, що додається.

### Опис основних змінних.

Глобальні (спільні для всіх процедур та функцій):

- Змінні *NValue*, *MValue* використовуються для задання вимірностей задачі на переставленнях; на розміщеннях використовуються змінні *NRValue*, *MRValue* та *MBRValue*.
- *PxR* – одновимірний масив, що містить всі елементи вектора *P* для задачі на розміщеннях.
- *Randomize* – вбудована процедура мови *Delphi*, ініціалізує генератор випадкових чисел (на основі системного годинника).
- *Px* – масив, що зберігає числа переставлення.
- *X* – масив, поточне обране переставлення в методі.
- *SumB*, *SumNB* – поточна сума та накопичена сума.
- *N\_v* – максимальний накопичений виграш.
- *Nv\_* – мінімальний накопичений програш.
- масив *BX* – скалярний добуток векторів.
- *\_v* – середнє накопиченого виграшу.
- *CmpEps* – значення точності порівняння чисел в алгоритмі.
- *AMatr* – програш першого гравця.
- *A* – платіжна матриця.
- *B* – стратегії другого гравця.
- *Sort2x* – процедура сортування двовимірної масиви за зростанням по першому стовпчику.

Змінні, що використовуються в функції *PrepareRozm*:

- *i* – поточне значення лічильника циклу.
- *aCurrInd* – номер поточного елемента в векторі *P* при генерації розміщення.
- *MaxNum* – поточне локальне значення вимірності вектора *P*.

- *aCurrNum* – поточна кількість елементів в генерованому розміщенні.
- *aGroupNum* – службова змінна, що вказує процедурі необхідну кількість елементів в розміщеннях.
- *aRozm* – масив, в якому зберігається генероване розміщення.
- *Result* – службова змінна мови *Delphi*, що містить вказівник результат функції.

Змінні, що використовуються в функції *GenerPx*:

- *iDelta* – поточний проміжок для генерації числа; використовується для досягнення необхідної умови рівності суми генерованого вектора одиниці.
- *i* – поточне значення лічильника циклу.
- *iTotalRange* – поточний модифікований загальний інтервал генерації, інтервал модифікується з метою кращої збіжності алгоритму генерації.
- *iStdRange* – зафіксований (стандартний) проміжок для генерації поточного випадкового числа в переставленні.
- *aCount* – кількість чисел в переставленні.
- *iCurRange* – поточний проміжок для генерації випадкового числа для переставлення.
- *iCurNum* – згенероване число.
- *arrTemp* – тимчасовий масив для збереження модифікованих (масштабованих) чисел в переставленні.
- *fPSum* – сума чисел в переставленні.

Змінні, що використовуються в функції *Eval\_B\_Part*:

- *i, j* – поточні значення лічильників циклу.
- *IterNum* – номер поточної ітерації.

Змінні, що використовуються в функції *Eval\_A\_Part*:

- *i, j* – поточні значення лічильників циклу.

- *MaxVal* – службова змінна для збереження поточного максимального числа.
- *MaxInd* – індекс поточного максимального елементу.
- *arrTemp* – службовий масив для збереження тимчасових даних.
- *StratCount* – масив, що зберігає частоту використання стратегій.
- *FindStrat* – функція пошуку стратегії серед використаних.
- *StInd* – індекс знайденої стратегії.
- *CurrStCount* – поточна кількість використаних стратегій.

### 3.3 Результати розрахунків та їх аналіз

За допомогою створеного програмного продукту існує можливість розв’язування задач на переставленнях, а також розміщеннях  $X = (x_1, \dots, x_m)$  за умов:  $\sum_{i=1}^m x_i = 1$ ,  $\sum_{i=1}^m x_i < 1$  та  $\sum_{i=1}^m x_i > 1$ .

Розглянемо розв’язування задачі з пункту 3.2 з використанням створеного програмного продукту. Нехай задано:

$$M = 10, m = 3, n = 2, P^x = \{0.1, 0.1, 0.3, 0.4, 0.5, 0.6, 0.7, 0.7, 0.9, 0.9\}, A = \begin{pmatrix} 7 & 3 \\ 1 & 5 \\ 3 & 4 \end{pmatrix}$$

Після введення початкових даних головне вікно програми матиме наступний вигляд (рис. 3.6).

Після натиснення на кнопку «Провести обчислення», ініціюється процес розв’язання введеної задачі. Для відображення детальних результатів позначимо прапорець «Вивід детальних таблиць». Після проведення обчислень отримаємо результат (рис. 3.7).

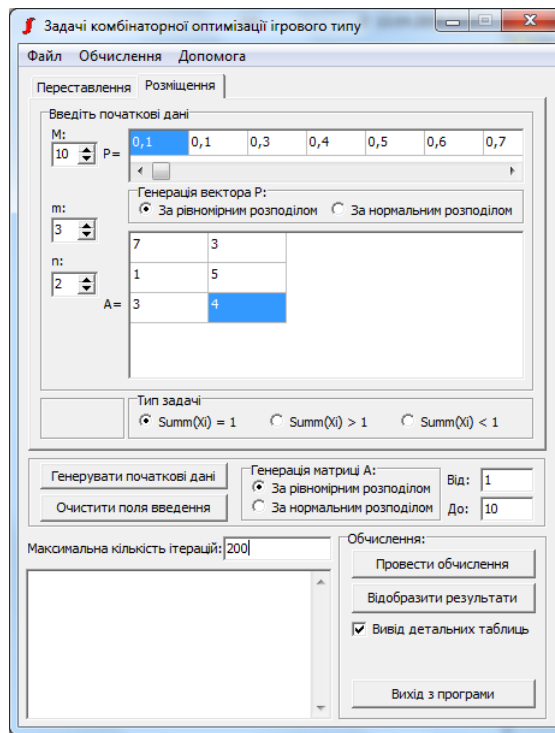


Рисунок 3.6 – Введення початкових даних

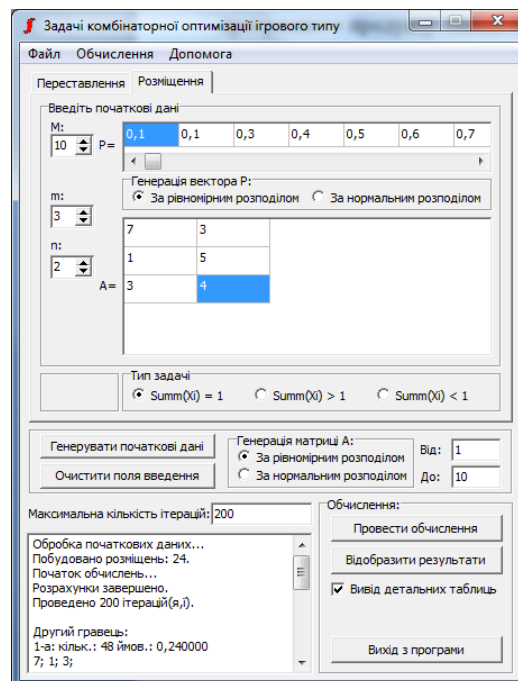


Рисунок 3.7 – Вікно програми з результати обчислень

Існує можливість перегляду детальних результатів обчислень, для цього необхідно натиснути кнопку «Відобразити результати». Відкриється наступне вікно програми (рис. 3.8):

| № | X   | B1 | B1X | B2 | B2X | Nv  | v   |
|---|-----|----|-----|----|-----|-----|-----|
| 1 | 0,4 | 7  | 2,8 | 3  | 1,2 |     |     |
|   | 0,9 | 1  | 0,9 | 5  | 4,5 |     |     |
|   | 0,7 | 3  | 2,1 | 4  | 2,8 |     |     |
|   | sum |    | 5,8 |    | 8,5 |     |     |
|   | SUM |    | 5,8 |    | 8,5 | 8,5 | 8,5 |
| 2 | 0,6 | 7  | 4,2 | 3  | 1,8 |     |     |
|   | 0,1 | 1  | 0,1 | 5  | 0,5 |     |     |
|   | 0,3 | 3  | 0,9 | 4  | 1,2 |     |     |
|   | sum |    | 5,2 |    | 3,5 |     |     |
|   | SUM |    | 11  |    | 12  | 12  | 6   |
| 3 | 0,6 | 7  | 4,2 | 3  | 1,8 |     |     |
|   | 0,1 | 1  | 0,1 | 5  | 0,5 |     |     |

Рисунок 3.8 – Детальні результати обчислень

В програмі, для подальшого аналізу та можливості друку результатів обчислень, реалізовано можливість експорту результатів в табличний процесор MS Excel. Для цього необхідно обрати пункт меню «Файл – Експорт результатів в Excel» форми результатів або «Обчислення – Експорт результатів в MS Excel» головної форми програми.

Для проведення досліджень на великій кількості задач існує можливість пакетного розрахунку задач, що генеруються з використанням вбудованого генератора задач. Для цього необхідно обрати пункт меню «Обчислення – Пакетний розрахунок». Для проведення обчислень необхідно ввести параметри генерованих задач в вікно пакетного розрахунку (рис. 3.9):

Рисунок 3.9 – Вікно пакетного розрахунку



Для прикладу розглянемо розв'язування 10 задач з параметрами  $M=10, m=5, n=5$  з максимальною кількістю ітерацій, рівну 150. Після проведення розрахунків (кнопка «Провести обчислення») отримаємо результат (рис. 3.10):

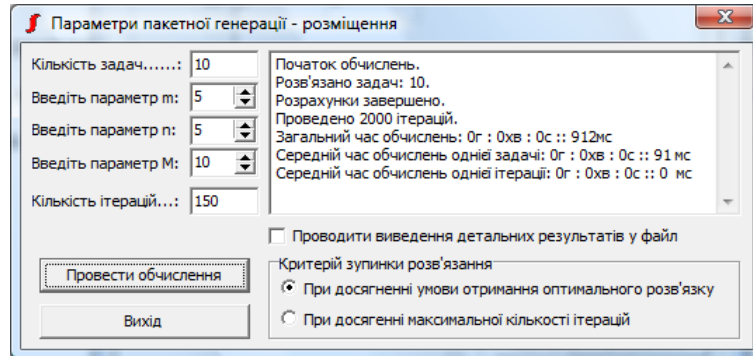


Рисунок 3.10 – Результати пакетного розрахунку

### 3.4 Інструкція по використанню програми

В рамках дипломної роботи було створено програму розв’язання задач комбінаторної оптимізації ігрового типу, в яких накладаються комбінаторні обмеження, що визначаються переставленнями та розміщеннями, на стратегії одного гравця. Її було реалізовано засобами мови програмування Delphi в середовищі програмування Embarcadero Delphi 10.4 Community Edition [12-14].

Основними властивостями даної програми є:

1. зручний користувальницький інтерфейс, особливістю якого є можливість введення початкових даних в табличному вигляді, збереження та завантаження їх із файлу, та відображення детальних результатів обчислень;
2. реалізовано експорт результатів обчислень в MS Excel;
3. в програмі реалізовано генератор задач, що дозволяє проводити дослідження часу розв’язання задач залежно від їх вимірності, а також протестувати роботу програми на великій кількості прикладів. Існує можливість генерувати випадкові числа за рівномірним або нормальним розподілом на вибір користувача.

Для запуску програми необхідно запустити \*.exe файл (на компакт-диску він знаходиться в папці «program/diplom.exe»).

Після запуску програми відкриється головна форма програми, яка має наступний вигляд (рис. 3.11).

Існує три можливості введення вихідних даних:

- безпосередньо з клавіатури;
- зчитування початкових даних з попередньо збереженого цією програмою файлу;
- генерація початкових даних.

Для роботи з програмою потрібно ввести початкові дані одним із перерахованих вище способом, або скористатися можливістю генерування початкових даних (натисненням кнопки «Генерувати дані»).

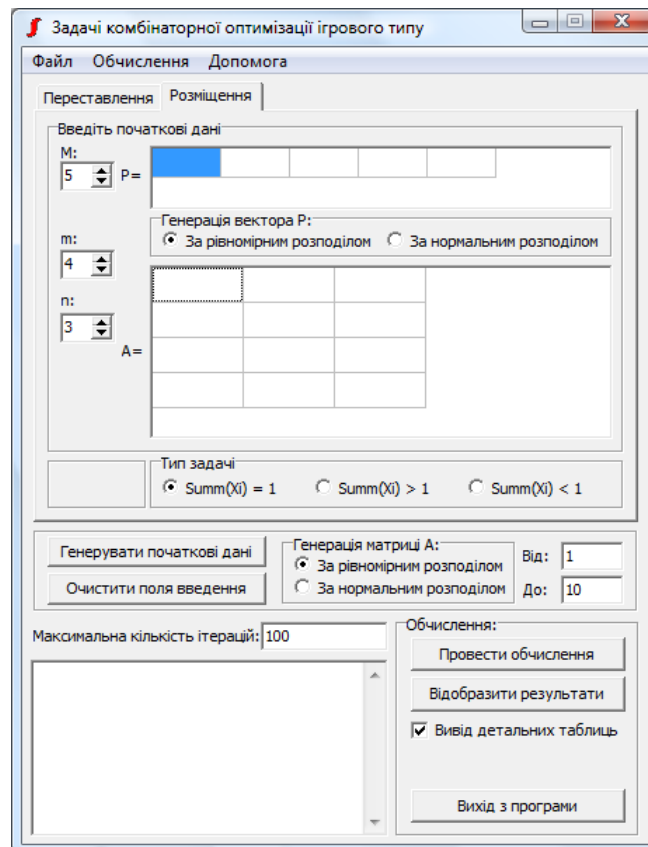


Рисунок 3.11 - Основне вікно програми

Для проведення обчислень задачі за введеними початковими даними необхідно натиснути кнопку «Провести обчислення». Результати роботи програми будуть відображені в текстовому полі на головному вікні програми.

При необхідності можливо переглянути детальні результати обчислень, натиснувши кнопку «Відобразити результати» (у випадку, коли встановлено прапорець «Вивід детальних таблиць»).

Всі вище згадані команди доступні також з головного меню програми.

## ВИСНОВКИ

В ході виконання дипломної роботи було розглянуто задачі комбінаторної оптимізації ігрового типу на переставленнях та розміщеннях. Головною метою було вивчення математичних моделей задач комбінаторної оптимізації ігрового типу на розміщеннях, розробка методу розв'язування та програмування алгоритму для таких задач.

Метод дослідження задач комбінаторної оптимізації ігрового типу, що використовуються в дипломній роботі, ґрунтується на теорії і методах евклідової комбінаторної оптимізації, лінійного програмування, алгебри та теорії ігор.

Основні результати роботи:

- вивчено клас задач, які поєднують в собі риси задач комбінаторної оптимізації та теорії ігор – задачі комбінаторної оптимізації ігрового типу, в яких на стратегії одного накладаються комбінаторні обмеження, що визначаються розміщеннями; побудовано моделі економічних задач, які зводяться до таких задач;
- розроблено ітераційний метод розв'язування задач комбінаторної оптимізації ігрового типу в яких на стратегії одного накладаються комбінаторні обмеження, що визначаються розміщеннями;
- розроблено алгоритм методу, який дозволяє знаходити оптимальну стратегію гравця, на стратегії якого накладаються комбінаторні обмеження в задачах комбінаторної оптимізації на розміщеннях, в яких комбінаторні обмеження накладаються тільки на стратегії одного гравця;
- програмно реалізовано даний метод розв'язання ігрових задач на стратегії якого накладаються комбінаторні обмеження, створено систему генерації задач з можливістю вибору типу розподілу при генерації;
- проведено тестування програми на великій кількості прикладів, досліджено залежність часу роботи алгоритму від вимірності задач за допомогою створеного інструменту пакетного розрахунку.

Практична ефективність запропонованого методу підтверджена числовими експериментами. Отримані результати дають підстави для того, щоб стверджувати, що результати обчислень, отримані за допомогою програми, є правильними. А отже, програма і сам метод є придатними для розв'язання цілого ряду практичних задач даного типу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Стоян Ю.Г., Ємець О.О. Теорія і методи евклідової комбінаторної оптимізації. – К.: ІСДО, 1993. – 188 с.
2. Ємець О. О., Барболіна Т. М. Властивості лінійних безумовних задач оптимізації на розміщеннях з імовірнісною невизначеністю. Доповіді НАН України. 2016. № 2. С. 31–37.
3. Ольховський Д. Розв'язування задач комбінаторної оптимізації ігрового типу на перестановках з обмеженнями на стратегії одного гравця / Д. Ольховський, О. Ольховська, О. Черненко, Т. Парфьонова, Ю. Олексійчук, О. Оріхівська, А. Задорожний // Інформаційні технології та суспільство, 2022. - Вип. 2 - С. 13-22.
4. О.О. Ємець, О.О. Черненко, Т.В. Чілікіна, О. В. Ольховська (2021). Огляд задач комбінаторної оптимізації визначення рентабельності сільськогосподарського виробництва та методи їх розв'язування Математичне та комп'ютерне моделювання. Серія: Фізико-математичні науки. Випуск 22, С. 63-74. 2021 р.
5. Стоян Ю.Г., Ємець О.О., Ємець Є.М. Оптимізація на полірозміщеннях: теорія та методи. – Полтава: РВЦ ПУСКУ, 2005. – 103 с.
6. Теорія ігор: курс лекцій [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалаври, за освітніми програмами «Системний аналіз і управління», «Системи і методи штучного інтелекту» спеціальностей 124 «Системний аналіз», 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського ; уклад. Л. В. Барановська. – Електронні текстові дані (1 файл: 21,06 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 245 с. – Назва з екрана.
7. Ємець О.О., Колечкіна Л.М. Задачі комбінаторної оптимізації з дробово-лінійними цільовими функціями:– К.: Наук. думка, 2005. – 117 с.
8. Ємець О.О., Роскладка О.В. Задачі оптимізації на полікомбінаторних множинах: властивості та розв'язування: – Полтава: РВЦ ПУСКУ, 2006. – 129 с.

9. Ємець О.О., Устьян Н.Ю. Ігрова комбінаторна модель однієї задачі сільськогосподарського виробництва – Матеріали VII Міжнародної науково-практичної конференції "Наука і освіта '2004" – Дніпропетровськ: Наука і освіта, 2004, том 70 – С.46-49.
10. Ємець О.О., Устьян Н.Ю. Один ітераційний метод розв'язування ігрових задач на переставленнях / О. О. Ємець, Н. Ю. Устьян // Наукові вісті НТУУ "КПІ". – 2008. – №3. – С.5-10.
11. Ємець О.О., Устьян Н.Ю. Моделювання і розв'язування деяких ігрових задач комбінаторної оптимізації економічного змісту – Збірник наукових праць: Економіка: Проблеми теорії і практики. Випуск 207. Том I – Дніпропетровськ: ДНУ, 2005 – С.82-99.
12. Embarcadero Delphi [Електронний ресурс]. - Режим доступу до ресурсу: URL: <https://www.embarcadero.com/products/delphi/>
13. Marco Cantu. Object Pascal Handbook. Delphi 10.4 Sydney Edition. The Complete Guide to the Object Pascal programming language for Delphi 10.4 Sydney. - 2021. pp. 44-49
14. Primož Gabrijelčič. Delphi High Performance. Packt Publishing. – 2018. – pp.122-131

## ДОДАТОК А

Таблиця А.1 Розв'язування ігрової задачі ітераційним методом

| $N$ | $X$ | $B_1$ | $B_1X$ | $B_2$ | $B_2X$ | $N\bar{v}$ | $\bar{v}$ | $j$   | $A_1$ | $A_2$ | $A_3$ | $N\underline{v}$ | $\underline{v}$ | $v^*$ |
|-----|-----|-------|--------|-------|--------|------------|-----------|-------|-------|-------|-------|------------------|-----------------|-------|
| 1   | 0,1 | 7     | 0,7    | 3     | 0,3    |            |           | 2     | 3     | 5     | 4     |                  |                 |       |
|     | 0,3 | 1     | 0,3    | 5     | 1,5    |            |           | sum   | 3     | 5     | 4     |                  |                 |       |
|     | 0,6 | 3     | 1,8    | 4     | 2,4    |            |           | SUM   | 3     | 5     | 4     | 3,5              | 3,5             | 3,85  |
|     | sum |       | 2,8    |       | 4,2    |            |           | NextX | 0,6   | 0,1   | 0,3   |                  |                 |       |
|     | SUM |       | 2,8    |       | 4,2    | 4,2        | 4,2       |       |       |       |       |                  |                 |       |
|     |     |       |        |       |        |            |           |       |       |       |       |                  |                 |       |
| 2   | 0,6 | 7     | 4,2    | 3     | 1,8    |            |           | 1     | 7     | 1     | 3     |                  |                 |       |
|     | 0,1 | 1     | 0,1    | 5     | 0,5    |            |           | sum   | 7     | 1     | 3     |                  |                 |       |
|     | 0,3 | 3     | 0,9    | 4     | 1,2    |            |           | SUM   | 10    | 6     | 7     | 6,7              | 3,35            | 3,675 |
|     | sum |       | 5,2    |       | 3,5    |            |           | NextX | 0,1   | 0,6   | 0,3   |                  |                 |       |
|     | SUM |       | 8      |       | 7,7    | 8          | 4         |       |       |       |       |                  |                 |       |
|     |     |       |        |       |        |            |           |       |       |       |       |                  |                 |       |
| 3   | 0,1 | 7     | 0,7    | 3     | 0,3    |            |           | 2     | 3     | 5     | 4     |                  |                 |       |
|     | 0,6 | 1     | 0,6    | 5     | 3      |            |           | sum   | 3     | 5     | 4     |                  |                 |       |
|     | 0,3 | 3     | 0,9    | 4     | 1,2    |            |           | SUM   | 13    | 11    | 11    | 11,2             | 3,73333         | 3,9   |
|     | sum |       | 2,2    |       | 4,5    |            |           | NextX | 0,1   | 0,3   | 0,6   |                  |                 |       |
|     | SUM |       | 10,2   |       | 12,2   | 12,2       | 4,06667   |       |       |       |       |                  |                 |       |
|     |     |       |        |       |        |            |           |       |       |       |       |                  |                 |       |
| 4   | 0,1 | 7     | 0,7    | 3     | 0,3    |            |           | 2     | 3     | 5     | 4     |                  |                 |       |
|     | 0,3 | 1     | 0,3    | 5     | 1,5    |            |           | sum   | 3     | 5     | 4     |                  |                 |       |
|     | 0,6 | 3     | 1,8    | 4     | 2,4    |            |           | SUM   | 16    | 16    | 15    | 15,4             | 3,85            | 3,975 |
|     | sum |       | 2,8    |       | 4,2    |            |           | NextX | 0,1   | 0,3   | 0,6   |                  |                 |       |
|     | SUM |       | 13     |       | 16,4   | 16,4       | 4,1       |       |       |       |       |                  |                 |       |
|     |     |       |        |       |        |            |           |       |       |       |       |                  |                 |       |
| 5   | 0,1 | 7     | 0,7    | 3     | 0,3    |            |           | 2     | 3     | 5     | 4     |                  |                 |       |
|     | 0,3 | 1     | 0,3    | 5     | 1,5    |            |           | sum   | 3     | 5     | 4     |                  |                 |       |
|     | 0,6 | 3     | 1,8    | 4     | 2,4    |            |           | SUM   | 19    | 21    | 19    | 19,2             | 3,84            | 3,98  |
|     | sum |       | 2,8    |       | 4,2    |            |           | NextX | 0,6   | 0,1   | 0,3   |                  |                 |       |
|     | SUM |       | 15,8   |       | 20,6   | 20,6       | 4,12      |       |       |       |       |                  |                 |       |
|     |     |       |        |       |        |            |           |       |       |       |       |                  |                 |       |
| 6   | 0,6 | 7     | 4,2    | 3     | 1,8    |            |           | 2     | 3     | 5     | 4     |                  |                 |       |
|     | 0,1 | 1     | 0,1    | 5     | 0,5    |            |           | sum   | 3     | 5     | 4     |                  |                 |       |
|     | 0,3 | 3     | 0,9    | 4     | 1,2    |            |           | SUM   | 22    | 26    | 23    | 22,7             | 3,78333         | 3,9   |
|     | sum |       | 5,2    |       | 3,5    |            |           | NextX | 0,6   | 0,1   | 0,3   |                  |                 |       |



Продовження таблиці А.1

|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
|----|-----|---|------|---|------|------|---------|-------|-----|-----|-----|------|---------|---------|
|    | SUM |   | 21   |   | 24,1 | 24,1 | 4,01667 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 7  | 0,6 | 7 | 4,2  | 3 | 1,8  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,1 | 1 | 0,1  | 5 | 0,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,3 | 3 | 0,9  | 4 | 1,2  |      |         | SUM   | 25  | 31  | 27  | 26,2 | 3,74286 | 3,84286 |
|    | sum |   | 5,2  |   | 3,5  |      |         | NextX | 0,6 | 0,1 | 0,3 |      |         |         |
|    | SUM |   | 26,2 |   | 27,6 | 27,6 | 3,94286 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 8  | 0,6 | 7 | 4,2  | 3 | 1,8  |      |         | 1     | 7   | 1   | 3   |      |         |         |
|    | 0,1 | 1 | 0,1  | 5 | 0,5  |      |         | sum   | 7   | 1   | 3   |      |         |         |
|    | 0,3 | 3 | 0,9  | 4 | 1,2  |      |         | SUM   | 32  | 32  | 30  | 30,8 | 3,85    | 3,8875  |
|    | sum |   | 5,2  |   | 3,5  |      |         | NextX | 0,1 | 0,3 | 0,6 |      |         |         |
|    | SUM |   | 31,4 |   | 31,1 | 31,4 | 3,925   |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 9  | 0,1 | 7 | 0,7  | 3 | 0,3  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,3 | 1 | 0,3  | 5 | 1,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 35  | 37  | 34  | 34,6 | 3,84444 | 3,88333 |
|    | sum |   | 2,8  |   | 4,2  |      |         | NextX | 0,3 | 0,1 | 0,6 |      |         |         |
|    | SUM |   | 34,2 |   | 35,3 | 35,3 | 3,92222 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 10 | 0,3 | 7 | 2,1  | 3 | 0,9  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,1 | 1 | 0,1  | 5 | 0,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 38  | 42  | 38  | 38,4 | 3,84    | 3,875   |
|    | sum |   | 4    |   | 3,8  |      |         | NextX | 0,6 | 0,1 | 0,3 |      |         |         |
|    | SUM |   | 38,2 |   | 39,1 | 39,1 | 3,91    |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 11 | 0,6 | 7 | 4,2  | 3 | 1,8  |      |         | 1     | 7   | 1   | 3   |      |         |         |
|    | 0,1 | 1 | 0,1  | 5 | 0,5  |      |         | sum   | 7   | 1   | 3   |      |         |         |
|    | 0,3 | 3 | 0,9  | 4 | 1,2  |      |         | SUM   | 45  | 43  | 41  | 42   | 3,81818 | 3,88182 |
|    | sum |   | 5,2  |   | 3,5  |      |         | NextX | 0,1 | 0,3 | 0,6 |      |         |         |
|    | SUM |   | 43,4 |   | 42,6 | 43,4 | 3,94545 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 12 | 0,1 | 7 | 0,7  | 3 | 0,3  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,3 | 1 | 0,3  | 5 | 1,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 48  | 48  | 45  | 46,2 | 3,85    | 3,875   |
|    | sum |   | 2,8  |   | 4,2  |      |         | NextX | 0,1 | 0,3 | 0,6 |      |         |         |
|    | SUM |   | 46,2 |   | 46,8 | 46,8 | 3,9     |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |

Продовження таблиці А.1

|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
|----|-----|---|------|---|------|------|---------|-------|-----|-----|-----|------|---------|---------|
| 13 | 0,1 | 7 | 0,7  | 3 | 0,3  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,3 | 1 | 0,3  | 5 | 1,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 51  | 53  | 49  | 50   | 3,84615 | 3,88462 |
|    | sum |   | 2,8  |   | 4,2  |      |         | NextX | 0,3 | 0,1 | 0,6 |      |         |         |
|    | SUM |   | 49   |   | 51   | 51   | 3,92308 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 14 | 0,3 | 7 | 2,1  | 3 | 0,9  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,1 | 1 | 0,1  | 5 | 0,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 54  | 58  | 53  | 53,8 | 3,84286 | 3,87857 |
|    | sum |   | 4    |   | 3,8  |      |         | NextX | 0,3 | 0,1 | 0,6 |      |         |         |
|    | SUM |   | 53   |   | 54,8 | 54,8 | 3,91429 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 15 | 0,3 | 7 | 2,1  | 3 | 0,9  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,1 | 1 | 0,1  | 5 | 0,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 57  | 63  | 57  | 57,6 | 3,84    | 3,87333 |
|    | sum |   | 4    |   | 3,8  |      |         | NextX | 0,5 | 0,1 | 0,4 |      |         |         |
|    | SUM |   | 57   |   | 58,6 | 58,6 | 3,90667 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 16 | 0,5 | 7 | 3,5  | 3 | 1,5  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,1 | 1 | 0,1  | 5 | 0,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,4 | 3 | 1,2  | 4 | 1,6  |      |         | SUM   | 60  | 68  | 61  | 61,1 | 3,81875 | 3,85312 |
|    | sum |   | 4,8  |   | 3,6  |      |         | NextX | 0,6 | 0,1 | 0,3 |      |         |         |
|    | SUM |   | 61,8 |   | 62,2 | 62,2 | 3,8875  |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 17 | 0,6 | 7 | 4,2  | 3 | 1,8  |      |         | 1     | 7   | 1   | 3   |      |         |         |
|    | 0,1 | 1 | 0,1  | 5 | 0,5  |      |         | sum   | 7   | 1   | 3   |      |         |         |
|    | 0,3 | 3 | 0,9  | 4 | 1,2  |      |         | SUM   | 67  | 69  | 64  | 65,4 | 3,84706 | 3,89412 |
|    | sum |   | 5,2  |   | 3,5  |      |         | NextX | 0,3 | 0,1 | 0,6 |      |         |         |
|    | SUM |   | 67   |   | 65,7 | 67   | 3,94118 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 18 | 0,3 | 7 | 2,1  | 3 | 0,9  |      |         | 1     | 7   | 1   | 3   |      |         |         |
|    | 0,1 | 1 | 0,1  | 5 | 0,5  |      |         | sum   | 7   | 1   | 3   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 74  | 70  | 67  | 68,6 | 3,81111 | 3,87778 |
|    | sum |   | 4    |   | 3,8  |      |         | NextX | 0,1 | 0,3 | 0,6 |      |         |         |
|    | SUM |   | 71   |   | 69,5 | 71   | 3,94444 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 19 | 0,1 | 7 | 0,7  | 3 | 0,3  |      |         | 1     | 7   | 1   | 3   |      |         |         |
|    | 0,3 | 1 | 0,3  | 5 | 1,5  |      |         | sum   | 7   | 1   | 3   |      |         |         |

Продовження таблиці А.1

|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
|----|-----|---|------|---|------|------|---------|-------|-----|-----|-----|------|---------|---------|
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 81  | 71  | 70  | 71,4 | 3,75789 | 3,82105 |
|    | sum |   | 2,8  |   | 4,2  |      |         | NextX | 0,1 | 0,3 | 0,6 |      |         |         |
|    | SUM |   | 73,8 |   | 73,7 | 73,8 | 3,88421 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 20 | 0,1 | 7 | 0,7  | 3 | 0,3  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,3 | 1 | 0,3  | 5 | 1,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 84  | 76  | 74  | 75,6 | 3,78    | 3,8375  |
|    | sum |   | 2,8  |   | 4,2  |      |         | NextX | 0,1 | 0,3 | 0,6 |      |         |         |
|    | SUM |   | 76,6 |   | 77,9 | 77,9 | 3,895   |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 21 | 0,1 | 7 | 0,7  | 3 | 0,3  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,3 | 1 | 0,3  | 5 | 1,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 87  | 81  | 78  | 79,8 | 3,8     | 3,85476 |
|    | sum |   | 2,8  |   | 4,2  |      |         | NextX | 0,1 | 0,3 | 0,6 |      |         |         |
|    | SUM |   | 79,4 |   | 82,1 | 82,1 | 3,90952 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 22 | 0,1 | 7 | 0,7  | 3 | 0,3  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,3 | 1 | 0,3  | 5 | 1,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 90  | 86  | 82  | 84   | 3,81818 | 3,87045 |
|    | sum |   | 2,8  |   | 4,2  |      |         | NextX | 0,1 | 0,3 | 0,6 |      |         |         |
|    | SUM |   | 82,2 |   | 86,3 | 86,3 | 3,92273 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 23 | 0,1 | 7 | 0,7  | 3 | 0,3  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,3 | 1 | 0,3  | 5 | 1,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 93  | 91  | 86  | 88,2 | 3,83478 | 3,88478 |
|    | sum |   | 2,8  |   | 4,2  |      |         | NextX | 0,1 | 0,3 | 0,6 |      |         |         |
|    | SUM |   | 85   |   | 90,5 | 90,5 | 3,93478 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 24 | 0,1 | 7 | 0,7  | 3 | 0,3  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,3 | 1 | 0,3  | 5 | 1,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 96  | 96  | 90  | 92,4 | 3,85    | 3,89792 |
|    | sum |   | 2,8  |   | 4,2  |      |         | NextX | 0,1 | 0,3 | 0,6 |      |         |         |
|    | SUM |   | 87,8 |   | 94,7 | 94,7 | 3,94583 |       |     |     |     |      |         |         |
|    |     |   |      |   |      |      |         |       |     |     |     |      |         |         |
| 25 | 0,1 | 7 | 0,7  | 3 | 0,3  |      |         | 2     | 3   | 5   | 4   |      |         |         |
|    | 0,3 | 1 | 0,3  | 5 | 1,5  |      |         | sum   | 3   | 5   | 4   |      |         |         |
|    | 0,6 | 3 | 1,8  | 4 | 2,4  |      |         | SUM   | 99  | 101 | 94  | 96,2 | 3,848   | 3,902   |
|    | sum |   | 2,8  |   | 4,2  |      |         | NextX | 0,3 | 0,1 | 0,6 |      |         |         |

Продовження таблиці А.1

|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
|----|-----|---|-------|---|-------|-------|---------|-------|-----|-----|-----|-------|---------|---------|
|    | SUM |   | 90,6  |   | 98,9  | 98,9  | 3,956   |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 26 | 0,3 | 7 | 2,1   | 3 | 0,9   |       |         | 2     | 3   | 5   | 4   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 3   | 5   | 4   |       |         |         |
|    | 0,6 | 3 | 1,8   | 4 | 2,4   |       |         | SUM   | 102 | 106 | 98  | 100   | 3,84615 | 3,89808 |
|    | sum |   | 4     |   | 3,8   |       |         | NextX | 0,3 | 0,1 | 0,6 |       |         |         |
|    | SUM |   | 94,6  |   | 102,7 | 102,7 | 3,95    |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 27 | 0,3 | 7 | 2,1   | 3 | 0,9   |       |         | 2     | 3   | 5   | 4   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 3   | 5   | 4   |       |         |         |
|    | 0,6 | 3 | 1,8   | 4 | 2,4   |       |         | SUM   | 105 | 111 | 102 | 103,8 | 3,84444 | 3,89444 |
|    | sum |   | 4     |   | 3,8   |       |         | NextX | 0,3 | 0,1 | 0,6 |       |         |         |
|    | SUM |   | 98,6  |   | 106,5 | 106,5 | 3,94444 |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 28 | 0,3 | 7 | 2,1   | 3 | 0,9   |       |         | 2     | 3   | 5   | 4   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 3   | 5   | 4   |       |         |         |
|    | 0,6 | 3 | 1,8   | 4 | 2,4   |       |         | SUM   | 108 | 116 | 106 | 107,6 | 3,84286 | 3,89107 |
|    | sum |   | 4     |   | 3,8   |       |         | NextX | 0,3 | 0,1 | 0,6 |       |         |         |
|    | SUM |   | 102,6 |   | 110,3 | 110,3 | 3,93929 |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 29 | 0,3 | 7 | 2,1   | 3 | 0,9   |       |         | 2     | 3   | 5   | 4   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 3   | 5   | 4   |       |         |         |
|    | 0,6 | 3 | 1,8   | 4 | 2,4   |       |         | SUM   | 111 | 121 | 110 | 111,4 | 3,84138 | 3,88793 |
|    | sum |   | 4     |   | 3,8   |       |         | NextX | 0,3 | 0,1 | 0,6 |       |         |         |
|    | SUM |   | 106,6 |   | 114,1 | 114,1 | 3,93448 |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 30 | 0,3 | 7 | 2,1   | 3 | 0,9   |       |         | 2     | 3   | 5   | 4   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 3   | 5   | 4   |       |         |         |
|    | 0,6 | 3 | 1,8   | 4 | 2,4   |       |         | SUM   | 114 | 126 | 114 | 115,2 | 3,84    | 3,885   |
|    | sum |   | 4     |   | 3,8   |       |         | NextX | 0,5 | 0,1 | 0,4 |       |         |         |
|    | SUM |   | 110,6 |   | 117,9 | 117,9 | 3,93    |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 31 | 0,5 | 7 | 3,5   | 3 | 1,5   |       |         | 2     | 3   | 5   | 4   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 3   | 5   | 4   |       |         |         |
|    | 0,4 | 3 | 1,2   | 4 | 1,6   |       |         | SUM   | 117 | 131 | 118 | 118,7 | 3,82903 | 3,87419 |
|    | sum |   | 4,8   |   | 3,6   |       |         | NextX | 0,6 | 0,1 | 0,3 |       |         |         |
|    | SUM |   | 115,4 |   | 121,5 | 121,5 | 3,91935 |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |

Продовження таблиці А.1

|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
|----|-----|---|-------|---|-------|-------|---------|-------|-----|-----|-----|-------|---------|---------|
| 32 | 0,6 | 7 | 4,2   | 3 | 1,8   |       |         | 2     | 3   | 5   | 4   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 3   | 5   | 4   |       |         |         |
|    | 0,3 | 3 | 0,9   | 4 | 1,2   |       |         | SUM   | 120 | 136 | 122 | 122,2 | 3,81875 | 3,8625  |
|    | sum |   | 5,2   |   | 3,5   |       |         | NextX | 0,6 | 0,1 | 0,3 |       |         |         |
|    | SUM |   | 120,6 |   | 125   | 125   | 3,90625 |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 33 | 0,6 | 7 | 4,2   | 3 | 1,8   |       |         | 2     | 3   | 5   | 4   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 3   | 5   | 4   |       |         |         |
|    | 0,3 | 3 | 0,9   | 4 | 1,2   |       |         | SUM   | 123 | 141 | 126 | 125,7 | 3,80909 | 3,85152 |
|    | sum |   | 5,2   |   | 3,5   |       |         | NextX | 0,6 | 0,1 | 0,3 |       |         |         |
|    | SUM |   | 125,8 |   | 128,5 | 128,5 | 3,89394 |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 34 | 0,6 | 7 | 4,2   | 3 | 1,8   |       |         | 2     | 3   | 5   | 4   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 3   | 5   | 4   |       |         |         |
|    | 0,3 | 3 | 0,9   | 4 | 1,2   |       |         | SUM   | 126 | 146 | 130 | 129,2 | 3,8     | 3,84118 |
|    | sum |   | 5,2   |   | 3,5   |       |         | NextX | 0,6 | 0,1 | 0,3 |       |         |         |
|    | SUM |   | 131   |   | 132   | 132   | 3,88235 |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 35 | 0,6 | 7 | 4,2   | 3 | 1,8   |       |         | 1     | 7   | 1   | 3   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 7   | 1   | 3   |       |         |         |
|    | 0,3 | 3 | 0,9   | 4 | 1,2   |       |         | SUM   | 133 | 147 | 133 | 134,4 | 3,84    | 3,86571 |
|    | sum |   | 5,2   |   | 3,5   |       |         | NextX | 0,3 | 0,1 | 0,6 |       |         |         |
|    | SUM |   | 136,2 |   | 135,5 | 136,2 | 3,89143 |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 36 | 0,3 | 7 | 2,1   | 3 | 0,9   |       |         | 1     | 7   | 1   | 3   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 7   | 1   | 3   |       |         |         |
|    | 0,6 | 3 | 1,8   | 4 | 2,4   |       |         | SUM   | 140 | 148 | 136 | 138,4 | 3,84444 | 3,86944 |
|    | sum |   | 4     |   | 3,8   |       |         | NextX | 0,3 | 0,1 | 0,6 |       |         |         |
|    | SUM |   | 140,2 |   | 139,3 | 140,2 | 3,89444 |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 37 | 0,3 | 7 | 2,1   | 3 | 0,9   |       |         | 1     | 7   | 1   | 3   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 7   | 1   | 3   |       |         |         |
|    | 0,6 | 3 | 1,8   | 4 | 2,4   |       |         | SUM   | 147 | 149 | 139 | 142,4 | 3,84865 | 3,87297 |
|    | sum |   | 4     |   | 3,8   |       |         | NextX | 0,3 | 0,1 | 0,6 |       |         |         |
|    | SUM |   | 144,2 |   | 143,1 | 144,2 | 3,8973  |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 38 | 0,3 | 7 | 2,1   | 3 | 0,9   |       |         | 1     | 7   | 1   | 3   |       |         |         |
|    | 0,1 | 1 | 0,1   | 5 | 0,5   |       |         | sum   | 7   | 1   | 3   |       |         |         |

# Закінчення таблиці А.1

|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
|----|-----|---|-------|---|-------|-------|---------|-------|-----|-----|-----|-------|---------|---------|
|    | 0,6 | 3 | 1,8   | 4 | 2,4   |       |         | SUM   | 154 | 150 | 142 | 145,6 | 3,83158 | 3,86579 |
|    | sum |   | 4     |   | 3,8   |       |         | NextX | 0,1 | 0,3 | 0,6 |       |         |         |
|    | SUM |   | 148,2 |   | 146,9 | 148,2 | 3,9     |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 39 | 0,1 | 7 | 0,7   | 3 | 0,3   |       |         | 2     | 3   | 5   | 4   |       |         |         |
|    | 0,3 | 1 | 0,3   | 5 | 1,5   |       |         | sum   | 3   | 5   | 4   |       |         |         |
|    | 0,6 | 3 | 1,8   | 4 | 2,4   |       |         | SUM   | 157 | 155 | 146 | 149,8 | 3,84103 | 3,85769 |
|    | sum |   | 2,8   |   | 4,2   |       |         | NextX | 0,1 | 0,3 | 0,6 |       |         |         |
|    | SUM |   | 151   |   | 151,1 | 151,1 | 3,87436 |       |     |     |     |       |         |         |
|    |     |   |       |   |       |       |         |       |     |     |     |       |         |         |
| 40 | 0,1 | 7 | 0,7   | 3 | 0,3   |       |         | 2     | 3   | 5   | 4   |       |         |         |
|    | 0,3 | 1 | 0,3   | 5 | 1,5   |       |         | sum   | 3   | 5   | 4   |       |         |         |
|    | 0,6 | 3 | 1,8   | 4 | 2,4   |       |         | SUM   | 160 | 160 | 150 | 154   | 3,85    | 3,86625 |
|    | sum |   | 2,8   |   | 4,2   |       |         | NextX | 0,1 | 0,3 | 0,6 |       |         |         |
|    | SUM |   | 153,8 |   | 155,3 | 155,3 | 3,8825  |       |     |     |     |       |         |         |