Міністерство освіти і науки України

Черкаський державний технологічний університет

Кафедра програмного забезпечення автоматизованих систем

**ЖУРНАЛ ЗВІТІВ**

З ЛАБОРАТОРНИХ РОБІТ

ПО ДИСЦИПЛІНІ «ТЕОРІЯ ПРИЙНЯТТЯ РІШЕНЬ»

|  |  |
| --- | --- |
| Перевірив роботи: | Виконав роботи: |
| «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | студент 1-го курсу гр. МПЗ-1904 |
| Професор каф. ПЗАС | Журавленко М.В. |
| Голуб С.В. |  |

Черкаси, 2020

**ЛАБОРАТОРНА РОБОТА №1**

**Тема роботи**: Аналіз альтернатив в умовах невизначеності.

**Мета роботи**: Засвоєння навичок оптимізації рішень в умовах невизначеності.  
 **Теоретичні відомості:** Більшість реальних задач прийняття рішень (ЗПР) містить в тому або іншому вигляді невизначеність. При використовуванні методів формалізації постановки і ухвалення рішень з урахуванням невизначеностей слід мати на увазі, що всі вони носять рекомендаційний характер і вибір остаточного рішення завжди залишається за особою, що приймає рішення (ОПР).

Коли немає інформації про ймовірності станів середовища (природи) для визначення найкращих рішень використовуються критерії максимаксу,

Для пошуку альтернативи за критерієм максимаксу потрібно відшукати найбільший елемент матриці (M). Номер рядка з цим елементом є номером альтернативи. Для визначення, чому дорівнює критерій Вальда (1.2), потрібно спочатку знайти найгірші значення у кожному рядку матриці показників (тобто найменші), зберегти їх з номерами рядка (і альтернативи), а потім з отриманих результатів обрати найбільше значення (W). Номер рядка з цим елементом є номером альтернативи.

Вираз для критерія Гурвіца вказує, що спочатку потрібно знайти найгірші значення у кожному рядку матриці показників (тобто найменші), зберегти їх з номерами рядка (і альтернативи), а потім – знайти найліпші значення у кожному рядку (тобто найбільші), зберегти їх з номерами рядка (і альтернативи). Отримаємо два стовпця. Далі у кожному рядку виконати множення найменшого а на q та найбільшого a на 1-q та сумування – результат зберегти з номером рядка. Отримаємо стовпець, операція max по якому дає найбільше значення (H A ), номер якого відповідає кращій альтернативі.

Для знаходження кращої альтернативи за критерієм Байєса-Лапласа матриця доповнюється ще одним рядком – рядком значень ймовірностей стану природи. Знаходиться нова матриця, для цього всі значення а ij потрібно помножити на р j . Значення кожного рядка підсумовуються. До сум застосовується max і номер того рядка, в якому є найбільша сума (Z BL ), дає кращу альтернативу.

**Код програми:**

**<?php**

**namespace App\DMT;**

**class Lab1 {**

**public $dmt\_spreadsheet = array();**

**public function \_\_construct($data)**

**{**

**$this->dmt\_spreadsheet = $data;**

**}**

**public function maxymax\_calculate($data, $best\_alternative = true) {**

**$val = $data[0][0];**

**$number\_of\_row\_of\_alternative = 0;**

**$result\_data = array();**

**foreach ($data as $index => $row) {**

**foreach ($row as $column) {**

**if ($best\_alternative) {**

**if ($column > $val) {**

**$val = $column;**

**$number\_of\_row\_of\_alternative = $index;**

**}**

**} else {**

**if ($column < $val) {**

**$val = $column;**

**$number\_of\_row\_of\_alternative = $index;**

**}**

**}**

**}**

**}**

**$result\_data["val"] = $val;**

**$result\_data["number\_of\_row\_of\_alternative"] = $number\_of\_row\_of\_alternative;**

**return $result\_data;**

**}**

**public function valda\_calculate($data, $best\_alternative = true) {**

**$val = -1;**

**$number\_of\_row\_of\_alternative = 0;**

**$result\_data = array();**

**$values\_from\_each\_row = array();**

**foreach ($data as $row) {**

**$suitable\_value\_from\_row = $row[0];**

**foreach ($row as $column) {**

**if ($best\_alternative) {**

**if ($column < $suitable\_value\_from\_row) $suitable\_value\_from\_row = $column;**

**} else {**

**if ($column > $suitable\_value\_from\_row) $suitable\_value\_from\_row = $column;**

**}**

**}**

**array\_push($values\_from\_each\_row, $suitable\_value\_from\_row);**

**}**

**$val = $values\_from\_each\_row[0];**

**foreach ($values\_from\_each\_row as $index => $suitable\_value) {**

**if ($best\_alternative) {**

**if ($suitable\_value > $val) {**

**$val = $suitable\_value;**

**$number\_of\_row\_of\_alternative = $index;**

**}**

**} else {**

**if ($suitable\_value < $val) {**

**$val = $suitable\_value;**

**$number\_of\_row\_of\_alternative = $index;**

**}**

**}**

**}**

**$result\_data["val"] = $val;**

**$result\_data["number\_of\_row\_of\_alternative"] = $number\_of\_row\_of\_alternative;**

**return $result\_data;**

**}**

**public function gurvitsa\_calculate($data, $q, $best\_alternative = true) {**

**$val = -1;**

**$number\_of\_row\_of\_alternative = 0;**

**$result\_data = array();**

**$min\_values\_from\_each\_row = array();**

**$max\_values\_from\_each\_row = array();**

**foreach ($data as $row) {**

**$suitable\_min\_value\_from\_row = $row[0];**

**$suitable\_max\_value\_from\_row = $row[0];**

**foreach ($row as $column) {**

**if ($best\_alternative) {**

**if ($column < $suitable\_min\_value\_from\_row) $suitable\_min\_value\_from\_row = $column;**

**if ($column > $suitable\_max\_value\_from\_row) $suitable\_max\_value\_from\_row = $column;**

**} else {**

**if ($column > $suitable\_min\_value\_from\_row) $suitable\_min\_value\_from\_row = $column;**

**if ($column < $suitable\_max\_value\_from\_row) $suitable\_max\_value\_from\_row = $column;**

**}**

**}**

**array\_push($min\_values\_from\_each\_row, $suitable\_min\_value\_from\_row);**

**array\_push($max\_values\_from\_each\_row, $suitable\_max\_value\_from\_row);**

**}**

**$val = $q \* $min\_values\_from\_each\_row[0] + (1 - $q) \* $max\_values\_from\_each\_row[0];**

**for ($i = 0; $i < count($min\_values\_from\_each\_row); $i++) {**

**$calculated\_values = $q \* $min\_values\_from\_each\_row[$i] + (1 - $q) \* $max\_values\_from\_each\_row[$i];**

**if ($best\_alternative) {**

**if ($calculated\_values > $val) {**

**$val = $calculated\_values;**

**$number\_of\_row\_of\_alternative = $i;**

**}**

**} else {**

**if ($calculated\_values < $val) {**

**$val = $calculated\_values;**

**$number\_of\_row\_of\_alternative = $i;**

**}**

**}**

**}**

**$result\_data["val"] = $val;**

**$result\_data["number\_of\_row\_of\_alternative"] = $number\_of\_row\_of\_alternative;**

**return $result\_data;**

**}**

**public function bayesa\_laplasa\_calculate($data, $probability, $best\_alternative = true) {**

**$val = -1;**

**$number\_of\_row\_of\_alternative = 0;**

**$result\_data = array();**

**$new\_data\_array = array();**

**foreach ($data as $row) {**

**$new\_row\_data = array();**

**foreach ($row as $index => $column) {**

**array\_push($new\_row\_data, $column \* $probability[$index]);**

**}**

**array\_push($new\_data\_array, $new\_row\_data);**

**}**

**$val = array\_sum($data[0]);**

**foreach ($new\_data\_array as $index => $row) {**

**if ($best\_alternative) {**

**if (array\_sum($row) > $val) {**

**$val = array\_sum($row);**

**$number\_of\_row\_of\_alternative = $index;**

**}**

**} else {**

**if (array\_sum($row) < $val) {**

**$val = array\_sum($row);**

**$number\_of\_row\_of\_alternative = $index;**

**}**

**}**

**}**

**$result\_data["val"] = $val;**

**$result\_data["number\_of\_row\_of\_alternative"] = $number\_of\_row\_of\_alternative;**

**return $result\_data;**

**}**

**//print maxymax\_calculate($data) . PHP\_EOL;**

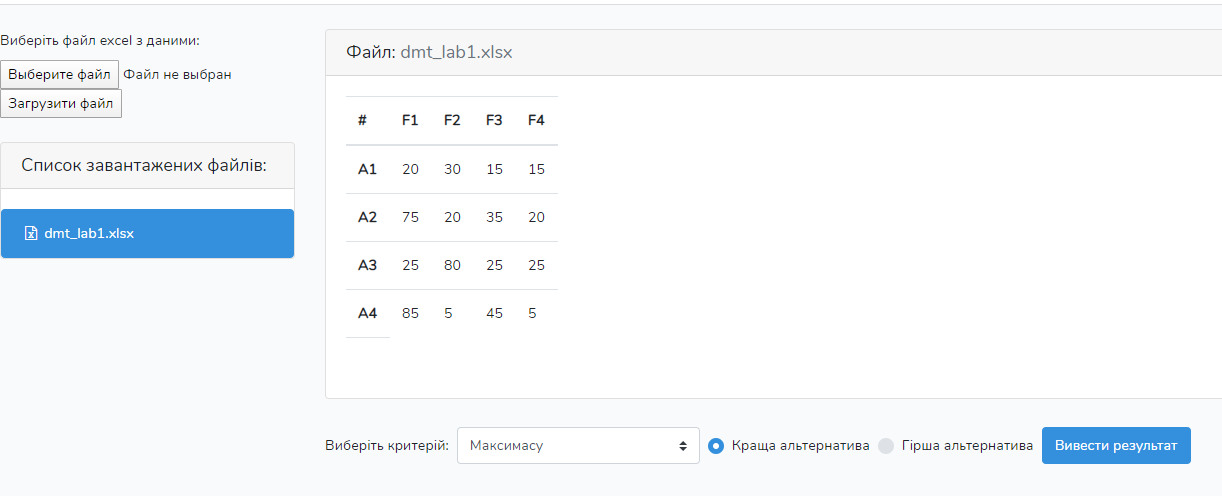
**//print valda\_calculate($data) . PHP\_EOL;**

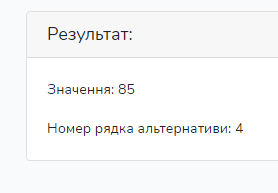
**//print gurvitsa\_calculate($data, 0.4) . PHP\_EOL;**

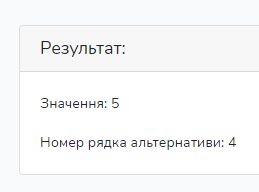
**//print bayesa\_laplasa\_calculate($data, array(0.7, 0.4, 0.5, 0.8));**

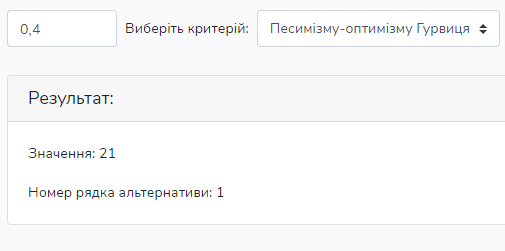
**}**

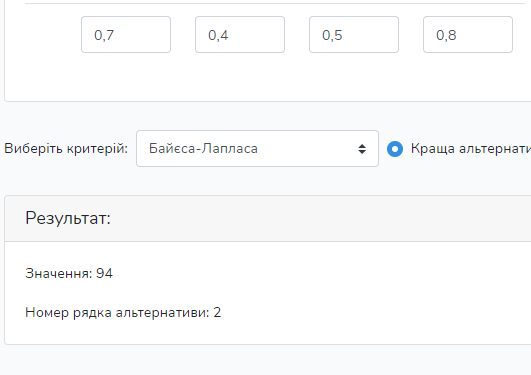
**Результат роботи програми:**

Для початку потрібно загрузити ексель файл з даними. І потім натиснути кнопку “Загрузити файл”. Потім появиться таблиця з даними . Знизу є список критеріїв, вибір кращої або гіршої альтернативи і кнопка “Вивести результат”.

Результат для критерію Максимаксу (краща альтернатива).

Результат для критерію Максимаксу (гірша альтернатива).

Результат для критерію Песимізму-оптимізму Гурвиця (краща альтернатива).

Результат для критерію Баєса-Лапласа (краща альтернатива).

**Висновок:** Під час проведення цієї лабораторної роботи я засвоїв навички оптимізації рішень в умовах невизначеності. Опанував критерій максимаксу, максимінний критерій Вальда, критерій песимізму-оптимізму Гурвиця, критерій Байєса-Лапласа.

**ЛАБОРАТОРНА РОБОТА №2**

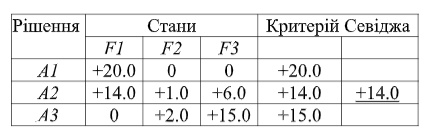
**Тема роботи**: Вибір рішення в умовах ризику.

**Мета роботи**: Засвоєння навичок аналізу альтернатив в умовах ризику.  
 **Теоретичні відомості:** Приклад для критерія Севіджа. При роботі комп’ютерів необхідно періодично припиняти обробку інформації і перевіряти на наявність вірусів. Зупинення призводить до певних економічних витрат. У разі ж якщо вірус вчасно виявлений не буде, можлива втрата деякої частини інформації, що призведе до ще більших збитків.

Альтернативи такі: А1– повна перевірка; А2 – мінімальна перевірка; А3 – відмова від перевірки. комп’ютерів може знаходитися в таких станах: F1 –

вірус відсутній; F2 – вірус є, але він не встиг пошкодити інформацію; F3 – є файли, які потребують відновлення. Результати, що включають витрати (в тисячах) на пошук вірусу і його ліквідацію, а також витрати, пов&#39;язані з відновленням інформації мають вигляд:

Матриця залишків для цього прикладу і їх оцінка згідно з критерієм Севіджа має вигляд:

Застосуємо критерій очікуваного значення у разі необхідності прийняти рішення про те, коли необхідно проводити профілактичний ремонт комп’ютерів, щоб мінімізувати втрати через несправність. У разі якщо ремонт проводитиметься занадто часто, витрати на обслуговування будуть великими при малих втратах через випадкову поломку. Так як неможливо передбачити заздалегідь, коли виникне несправність, необхідно знайти ймовірність того, що комп’ютерів вийде з ладу в період часу t. У цьому і полягає елемент ризику. Математично це виглядає так: комп’ютерів ремонтується індивідуально, якщо вона зупинилася через поломку. Через T інтервалів часу виконується профілактичний ремонт всіх n комп’ютерів. Необхідно визначити оптимальне значення Т, при якому мінімізуються загальні витрати на ремонт несправних комп’ютерів і проведення профілактичного ремонту в розрахунку на один інтервал часу.

Нехай р t – ймовірність виходу з ладу однієї комп’ютерів в момент t, а n t –

випадкове значення кількості всіх комп’ютерів, що вийшли з ладу в той же

момент. Нехай далі С 1 – витрати на ремонт несправної комп’ютерів і С 2 –

витрати на профілактичний ремонт однієї машини. Застосування критерію очікуваного значення в даному випадку виправдано, якщо комп’ютери працюють впродовж тривалого проміжку часу. При цьому очікувані витрати на один інтервал складуть, де M(n t ) – математичне сподівання кількості комп’ютерів, що вийшли з ладу в момент t. Комп’ютерів. Так як n t має біноміальний розподіл з параметрами (n, p t ), то M(n t ) = np t . Тому очікувані витрати ОВ.

Необхідні умови оптимальності T\* мають вигляд:

ОВ (T\*-1) = ОВ (T\*),

ОВ (T\*+1) =ОВ (T\*).

Отже, починаючи з малого значення T, обчислюють ОВ (T), поки не

будуть задоволені необхідні умови оптимальності.

Нехай С 1 = 100; С 2 = 10; n = 50. Значення p t мають вигляд:

**Код програми:**

**<?php**

**namespace App\DMT;**

**class Lab2 {**

**public $dmt\_spreadsheet = array();**

**public function \_\_construct($data)**

**{**

**$this->dmt\_spreadsheet = $data;**

**}**

**public function savage\_calculate($data, $best\_alternative = true) {**

**$number\_of\_row\_of\_alternative = 0;**

**$result\_data = array();**

**$risk\_data\_array = array();**

**$max\_values\_from\_each\_column = $data[0];**

**$max\_values\_from\_each\_row = array();**

**foreach ($data as $row) {**

**foreach ($row as $index => $column) {**

**if ($column > $max\_values\_from\_each\_column[$index]) {**

**$max\_values\_from\_each\_column[$index] = $column;**

**}**

**}**

**}**

**foreach ($data as $row) {**

**$risk\_row\_data = array();**

**foreach ($row as $index => $column) {**

**array\_push($risk\_row\_data, $max\_values\_from\_each\_column[$index] - $column);**

**}**

**array\_push($risk\_data\_array, $risk\_row\_data);**

**}**

**foreach ($risk\_data\_array as $row) {**

**$suitable\_max\_values\_from\_row = $row[0];**

**foreach ($row as $column) {**

**if ($column > $suitable\_max\_values\_from\_row) $suitable\_max\_values\_from\_row = $column;**

**}**

**array\_push($max\_values\_from\_each\_row, $suitable\_max\_values\_from\_row);**

**}**

**$val = $max\_values\_from\_each\_row[0];**

**foreach ($max\_values\_from\_each\_row as $index => $max\_value) {**

**if ($best\_alternative) {**

**if ($max\_value < $val) {**

**$val = $max\_value;**

**$number\_of\_row\_of\_alternative = $index;**

**}**

**} else {**

**if ($max\_value > $val) {**

**$val = $max\_value;**

**$number\_of\_row\_of\_alternative = $index;**

**}**

**}**

**}**

**$result\_data["val"] = $val;**

**$result\_data["number\_of\_row\_of\_alternative"] = $number\_of\_row\_of\_alternative;**

**$result\_data["risk\_data\_array"] = $risk\_data\_array;**

**return $result\_data;**

**}**

**public function expected\_value\_calculate($data, $c1, $c2, $n) {**

**$number\_of\_row\_of\_alternative = 0;**

**$result\_data = array();**

**foreach ($data as $index => $row) {**

**$p\_sum = 0;**

**if ($index != 0) {**

**for ($i = 0; $i < $index; $i++) {**

**$p\_sum += $data[$i][0];**

**}**

**}**

**array\_push($data[$index], $p\_sum);**

**array\_push($data[$index], round($n \* ($c1 \* $p\_sum + $c2) / ($index + 1), 1));**

**}**

**$val = 0;**

**foreach ($data as $index => $row) {**

**if ($index != 0) {**

**if (($data[$index - 1][2] >= $data[$index][2]) && ($data[$index + 1][2] >= $data[$index][2])) {**

**$val = $data[$index][2];**

**$number\_of\_row\_of\_alternative = $index;**

**}**

**}**

**}**

**$result\_data["val"] = $val;**

**$result\_data["number\_of\_row\_of\_alternative"] = $number\_of\_row\_of\_alternative;**

**$result\_data["new\_data"] = $data;**

**return $result\_data;**

**}**

**public function expected\_value\_dispersion\_calculate($data, $c1, $c2, $n) {**

**$number\_of\_row\_of\_alternative = 0;**

**$result\_data = array();**

**foreach ($data as $index => $row) {**

**array\_push($data[$index], pow($data[$index][0], 2));**

**$p\_sum = 0;**

**$p\_pow\_2\_sum = 0;**

**if ($index != 0) {**

**for ($i = 0; $i < $index; $i++) {**

**$p\_sum += $data[$i][0];**

**}**

**}**

**if ($index != 0) {**

**for ($i = 0; $i < $index; $i++) {**

**$p\_pow\_2\_sum += $data[$i][1];**

**}**

**}**

**array\_push($data[$index], $p\_sum);**

**array\_push($data[$index], $p\_pow\_2\_sum);**

**array\_push($data[$index], round($n / ($index + 1) \* ($c1 \* $data[$index][2] + $c2) + $n \* pow($c1 / ($index + 1), 2) \* ( $data[$index][2] - $data[$index][3]), 2));**

**}**

**$val = $data[0][4];**

**foreach ($data as $index => $row) {**

**if ($row[4] < $val) {**

**$val = $row[4];**

**$number\_of\_row\_of\_alternative = $index;**

**}**

**}**

**$result\_data["val"] = $val;**

**$result\_data["number\_of\_row\_of\_alternative"] = $number\_of\_row\_of\_alternative;**

**$result\_data["new\_data"] = $data;**

**return $result\_data;**

**}**

**public function average\_expected\_value\_calculate($data, $best\_alternative = true) {**

**$number\_of\_row\_of\_alternative = 0;**

**$result\_data = array();**

**$cases\_count\_array = array();**

**$average\_expected\_value\_array = array();**

**foreach ($data as $row) {**

**$cases\_count = 0;**

**foreach ($row as $column) {**

**$cases\_count += $column[1];**

**}**

**array\_push($cases\_count\_array, $cases\_count);**

**}**

**foreach ($data as $index => $row) {**

**$average\_expected\_value = 0;**

**foreach ($row as $index2 => $column) {**

**$average\_expected\_value += $column[0] \* $column[1] / $cases\_count\_array[$index];**

**$data[$index][$index2][2] = $column[1] / $cases\_count\_array[$index];**

**}**

**array\_push($average\_expected\_value\_array, $average\_expected\_value);**

**}**

**$val = $average\_expected\_value\_array[0];**

**foreach ($average\_expected\_value\_array as $index => $expected\_value) {**

**if ($best\_alternative) {**

**if ($expected\_value > $val) {**

**$val = $expected\_value;**

**$number\_of\_row\_of\_alternative = $index;**

**}**

**} else {**

**if ($expected\_value < $val) {**

**$val = $expected\_value;**

**$number\_of\_row\_of\_alternative = $index;**

**}**

**}**

**}**

**$result\_data["val"] = $val;**

**$result\_data["number\_of\_row\_of\_alternative"] = $number\_of\_row\_of\_alternative;**

**$result\_data["average\_expected\_value\_array"] = $average\_expected\_value\_array;**

**$result\_data["new\_data"] = $data;**

**return $result\_data;**

**}**

**public function limit\_level\_calculate($i1, $i2, $a1, $a2) {**

**$result\_data = array();**

**$expected\_deficit = array();**

**$expected\_surpluses = array();**

**$expected\_deficit\_limit = round(log($i2) - ($a1 / $i2) - 1, 3);**

**$expected\_surpluses\_limit = round(log($i1) - ($a2 / $i2) - 1, 3);**

**for ($i = $i1; $i <= $i2; $i++) {**

**array\_push($expected\_deficit, round(log($i) - ($i / $i2), 2));**

**array\_push($expected\_surpluses, round(log($i) - ($i / $i1), 2));**

**}**

**for ($i = 0; $i < count($expected\_deficit); $i++) {**

**if ($expected\_deficit[$i] >= $expected\_deficit\_limit && $expected\_surpluses[$i] >= $expected\_surpluses\_limit) {**

**$suitable\_start\_interval\_index = $i;**

**break;**

**}**

**}**

**for ($i = count($expected\_deficit) - 1; $i >= 0; $i--) {**

**if ($expected\_deficit[$i] >= $expected\_deficit\_limit && $expected\_surpluses[$i] >= $expected\_surpluses\_limit) {**

**$suitable\_end\_interval\_index = $i;**

**break;**

**}**

**}**

**$result\_data["expected\_deficit"] = $expected\_deficit;**

**$result\_data["expected\_surpluses"] = $expected\_surpluses;**

**$result\_data["expected\_deficit\_limit"] = $expected\_deficit\_limit;**

**$result\_data["expected\_surpluses\_limit"] = $expected\_surpluses\_limit;**

**$result\_data["suitable\_start\_interval\_index"] = $suitable\_start\_interval\_index;**

**$result\_data["suitable\_end\_interval\_index"] = $suitable\_end\_interval\_index;**

**return $result\_data;**

**}**

**//print savage\_calculate($data)["val"] . PHP\_EOL;**

**//print expected\_value\_calculate($data2, 100, 10, 50)["val"] . PHP\_EOL;**

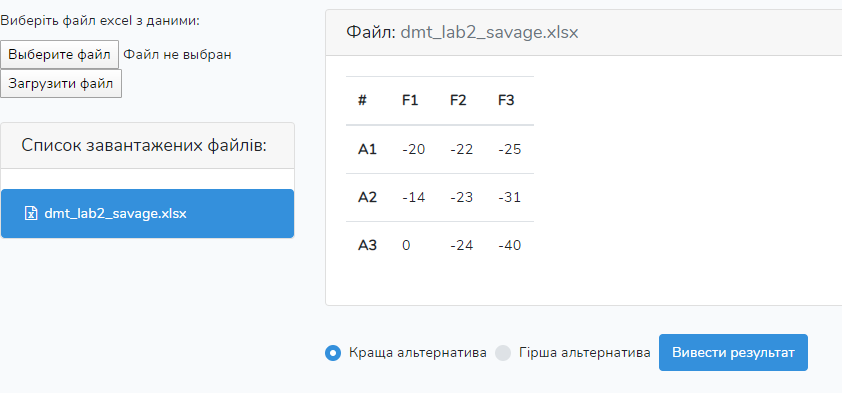
**//print expected\_value\_dispersion\_calculate($data2, 100, 10, 50)["val"] . PHP\_EOL;**

**//print average\_expected\_value\_calculate($data3)["val"] . PHP\_EOL;**

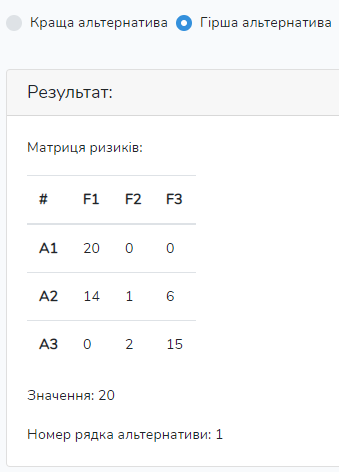
**//print limit\_level\_calculate(10, 20, 2, 4) . PHP\_EOL;**

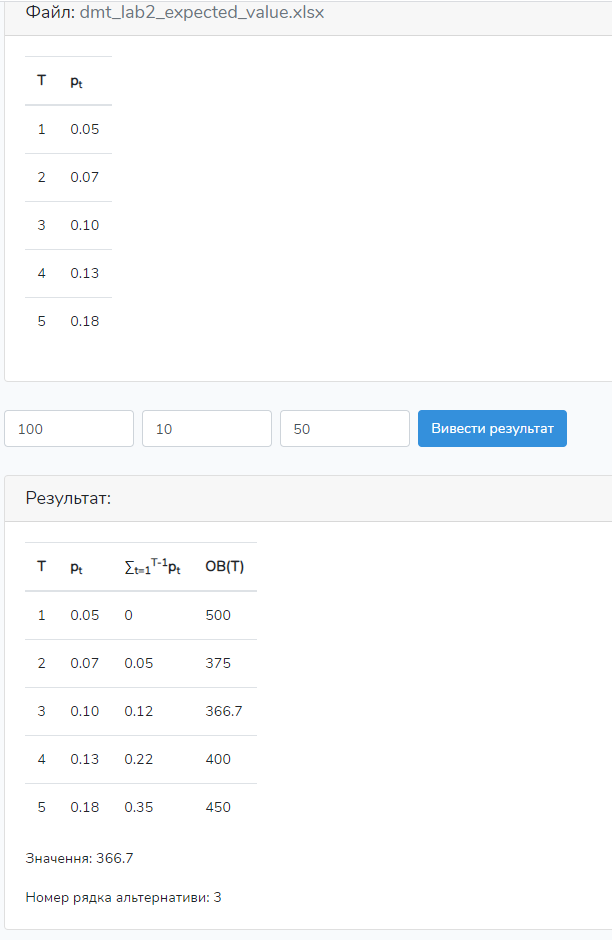
**}**

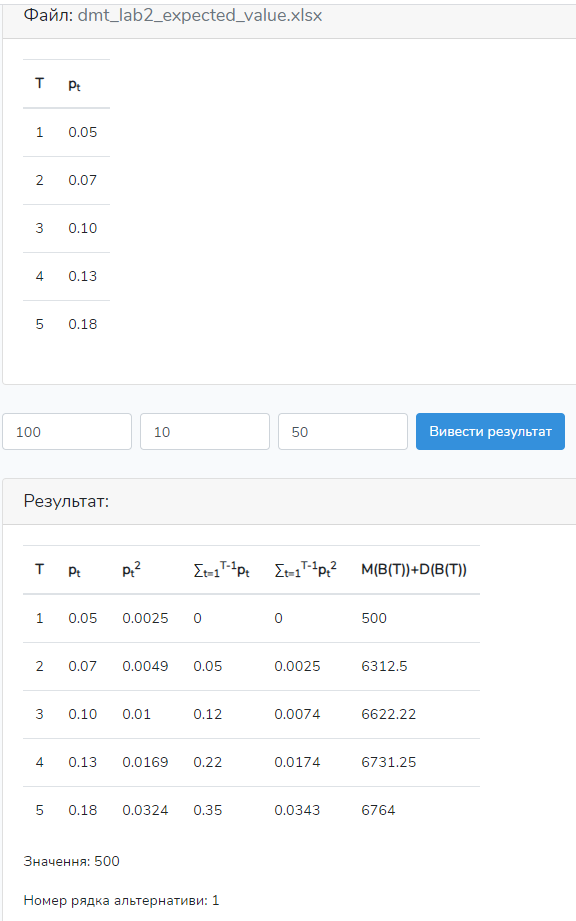
**Результат роботи програми:**

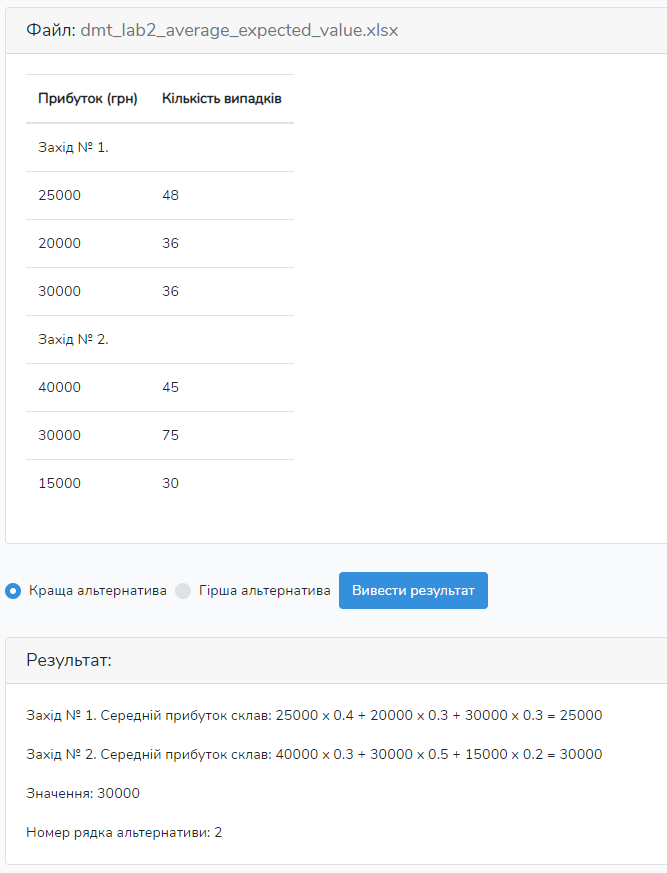
Для початку потрібно загрузити ексель файл з даними. І потім натиснути кнопку “Загрузити файл”. Потім появиться таблиця з даними . Знизу є список критеріїв, вибір кращої або гіршої альтернативи і кнопка “Вивести результат”.

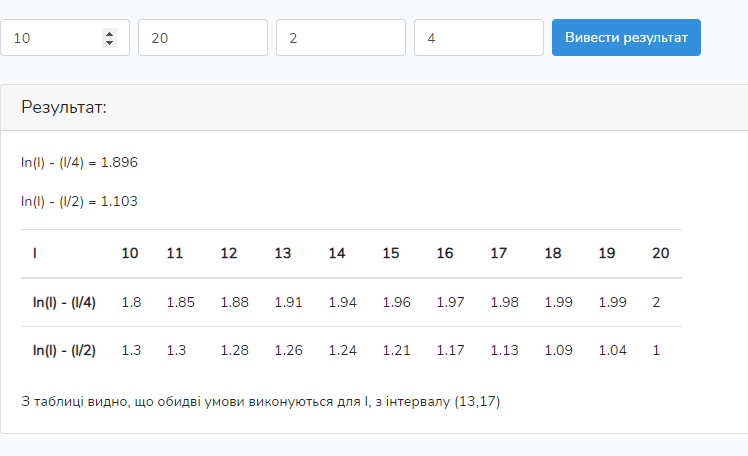
Результат для критерію Севіджа (краща альтернатива).

Результат для критерію Севіджа (гірша альтернатива).

Результат для критерію очікуваного значення.

Результат для критерію комбінації очікуваного значення і дисперсії.

Результат для критерію середнього очікуваного значення.

Результат для критерію граничного рівня.

**Висновок:** Під час проведення цієї лабораторної роботи я засвоїв навички аналізу альтернатив в умовах ризику. Опанував критерій Севіджа, критерій очікуваного значення, критерій комбінації очікуваного значення і дисперсії, критерій відомого граничного рівня, критерій найвірогіднішої події в майбутньому.

**ЛАБОРАТОРНА РОБОТА №3**

**Тема:** Застосування методу аналізу при прийнятті групових рішень.

**Мета:** Закріпити навички використання методу аналізу при прийнятті групових рішень.

**Теоретичні відомості:** Прийняття рішень ґрунтується на порівнянні кількох можливих альтернативних рішень та виборі одного рішення, яке найбільш відповідає висунутим критеріям. Дана робота дозволяє закріпити навички використання системного підходу при організації роботи групи експертів з пошуку можливих варіантів рішень.

Алгоритм пошуку:

1. Формулювання проблеми. Визначення змісту проблеми.

2. Визначення причин, які призводять до виникнення даної проблеми (10- 12 пунктів).

3. Визначення всіх можливих шляхів усунення кожної із причин сформульованої проблеми (10-12 пунктів на кожну причину).

4. Визначення прийнятних шляхів усунення кожної причини.

5. Складання мережевого графіку виконання робіт із усунення даної

проблеми.

**Код програми:**

**<?php**

**namespace App\DMT;**

**class Lab3 {**

**public $dmt\_spreadsheet = array();**

**public function \_\_construct($data)**

**{**

**$this->dmt\_spreadsheet = $data;**

**}**

**public function summation\_of\_ranks\_calculate($data, $best\_alternative = true) {**

**$result\_data = array();**

**$ranks\_array = array();**

**foreach ($data as $row) {**

**foreach ($row as $index => $column) {**

**if ($index != 0) {**

**if (!isset($ranks\_array[$column])) {**

**$ranks\_array[$column] = array();**

**}**

**array\_push($ranks\_array[$column], $index);**

**}**

**}**

**}**

**$val = array\_sum(array\_values($ranks\_array)[0]);**

**$alternative = array\_keys($ranks\_array)[0];**

**foreach ($ranks\_array as $index => $rank) {**

**if ($best\_alternative) {**

**if (array\_sum($rank) < $val) {**

**$val = array\_sum($rank);**

**$alternative = $index;**

**}**

**} else {**

**if (array\_sum($rank) > $val) {**

**$val = array\_sum($rank);**

**$alternative = $index;**

**}**

**}**

**}**

**$result\_data["val"] = $val;**

**$result\_data["alternative"] = $alternative;**

**$result\_data["ranks\_array"] = $ranks\_array;**

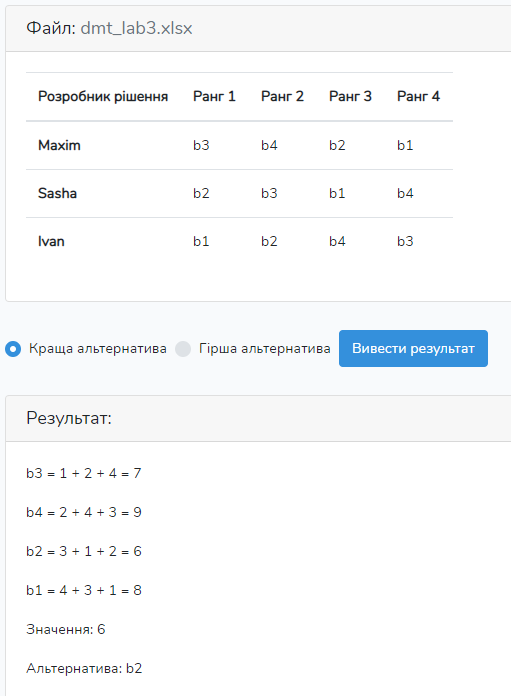
**return $result\_data;**

**}**

**//print summation\_of\_ranks\_calculate($data)["val"] . PHP\_EOL;**

**}**

**Результат роботи програми:**

Результат для методу аналізу (краща альтернатива).

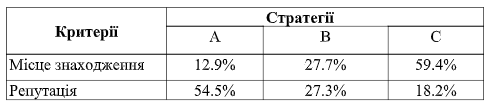
**Висновок:** Під час проведення цієї лабораторної роботи я засвоїв навички використання методу аналізу при прийнятті групових рішень. Дослідив роботу цього методу на даних з ексель файлу.

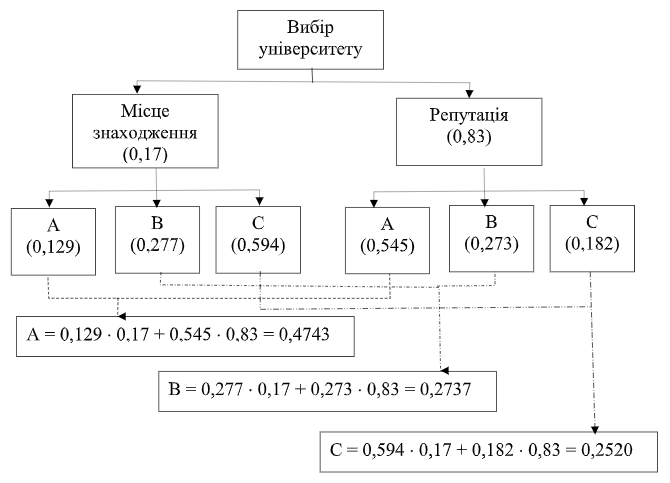
**ЛАБОРАТОРНА РОБОТА №4**

**Тема:** Використання методу аналізу ієрархій для вирішення задач системного аналізу.

**Мета:** Сформувати навички розпізнавання типів задач системного аналізу, які можуть бути вирішені з використанням методу аналізу ієрархій, які мають один або кілька критеріальних рівні.

**Теоретичні відомості:** Мартін Ганс — випускник-відмінник середньої школи, що одержав повну стипендію від трьох університетів: А, В и С. З метою вибору університету Мартін сформулював два основних критерії: місцезнаходження університету і його академічна репутація. Будучи відмінним учнем, він оцінює академічну репутацію університету в п’ять разів вище, ніж його місцезнаходження. Це приводить до того, що репутації університету приписується вага приблизно 83%, а його місцезнаходженню — 17%. Далі Мартін використовує системний аналіз (сутність його викладається нижче) для оцінки трьох університетів з погляду їхній місцезнаходження і репутації. Проведень аналіз дає наступні оцінки.

Структура задачі прийняття рішень приведена на рис. 4.1. Задача має єдиний ієрархічний рівень із двома критеріями (місцезнаходження і репутація) і три альтернативних рішення (університети А, В и С).



Оцінка трьох університетів заснована на обчисленні комбінованого вагового коефіцієнта для кожного з них.

Університет А: 0,17·0,129+0,83·0,545=0,4743

Університет У: 0,17·0,277+0,83·0,273=0,2737

Університет З: 0,17·0,594+0,83·0,182=0,2520

На основі цих обчислень університет А одержує найвищу комбіновану вагу і, отже, є найбільш оптимальним вибором Мартіна.

Загальна структура методу аналізу ієрархій може включати кілька ієрархічних рівнів із своїми критеріями. Припустимо, що в для умов попереднього прикладу, сестра-близнюк Мартіна Джейн також одержала повну стипендію від трьох університетів. Однак і батьки ставлять умову, що діти повинні учитися в одному університеті, тоді вони зможуть користатися одним автомобілем. На рис. 4.2 приведена структура задачі вибору рішення, що тепер включає два ієрархічних рівні зі своїми критеріями. Величини p і q (приблизно рівні) на першому ієрархічному рівні представляють собою вагові коефіцієнти, що приписуються точці зору Мартіна і Джейн щодо процесу вибору, відповідно. Другий ієрархічний рівень використовує ваги (р 1 ,р 2 ) і (q 1 ,q 2 ) для відображення індивідуальних точок зору Мартіна і Джейн щодо критеріїв місцезнаходження й академічної репутації кожного університету. Інша частина структури ухвалення рішення може бути інтерпретована аналогічно попередньому прикладу.

Складність методу аналізу ієрархій полягає у визначенні відносних вагових коефіцієнтів для оцінки альтернативних рішень. Якщо мається п критеріїв на заданому рівні ієрархії, що відповідає процедура створює матрицю А розмірності n × п, іменовану матрицею парних порівнянь, що відбиває судження обличчя, що приймає рішення, щодо важливості різних критеріїв. Парне порівняння виконаються таким чином, що критерій у рядку і (і= 1, 2,..., п) оцінюється щодо кожного з критеріїв, представлених п стовпцями. Позначимо через a ij елемент матриці А, що знаходиться на перетинанні i-й рядка j-го стовпця. Відповідно до методу аналізу ієрархій для опису згаданих оцінок використовуються цілі числа від 1 до 9. При цьому a ij = 1 означає, что i-й і j-й критерії однаково важливі, а ij = 5 відбиває думку, що i-й ритерій значно важливіше, ніж j-й, а а ij = 9 указує, що i-й критерій набагато важливіше j  го. Інші проміжні значення між 1 і 9 інтерпретуються аналогічно.

Погодженість таких позначень забезпечується наступним умовою: якщо a ij = k, то автоматично a ji = 1/k. Крім того, усі діагональні елементи а ii матриці А повинні бути рівні 1, тому що вони виражають оцінку критерію щодо самих себе.

**Код програми:**

**<?php**

**namespace App\DMT;**

**class Lab4 {**

**public $dmt\_spreadsheet = array();**

**public function \_\_construct($data)**

**{**

**$this->dmt\_spreadsheet = $data;**

**}**

**public function analysis\_of\_hierarchies\_calculate($data, $best\_alternative = true) {**

**$result\_data = array();**

**$strategy\_array = array();**

**$strategy\_sum = array();**

**foreach ($data as $key => $row) {**

**foreach ($row as $index => $column) {**

**if ($index > 1) {**

**if (!isset($strategy\_array[$index - 2])) {**

**$strategy\_array[$index - 2] = array();**

**}**

**$addition\_element = array();**

**array\_push($addition\_element, $column/100);**

**array\_push($addition\_element, $data[$key][1]/100);**

**array\_push($strategy\_array[$index - 2], $addition\_element);**

**}**

**}**

**}**

**$val = 0;**

**$alternative = 0;**

**foreach ($strategy\_array as $index => $strategy) {**

**$addition\_sum = 0;**

**foreach ($strategy as $addition\_element) {**

**$addition\_sum += $addition\_element[0] \* $addition\_element[1];**

**}**

**array\_push($strategy\_sum, $addition\_sum);**

**if ($index == 0) {**

**$val = $addition\_sum;**

**}**

**if ($best\_alternative) {**

**if ($addition\_sum > $val) {**

**$val = $addition\_sum;**

**$alternative = $index;**

**}**

**} else {**

**if ($addition\_sum < $val) {**

**$val = $addition\_sum;**

**$alternative = $index;**

**}**

**}**

**}**

**$result\_data["val"] = $val;**

**$result\_data["alternative"] = $alternative;**

**$result\_data["strategy\_array"] = $strategy\_array;**

**$result\_data["strategy\_sum"] = $strategy\_sum;**

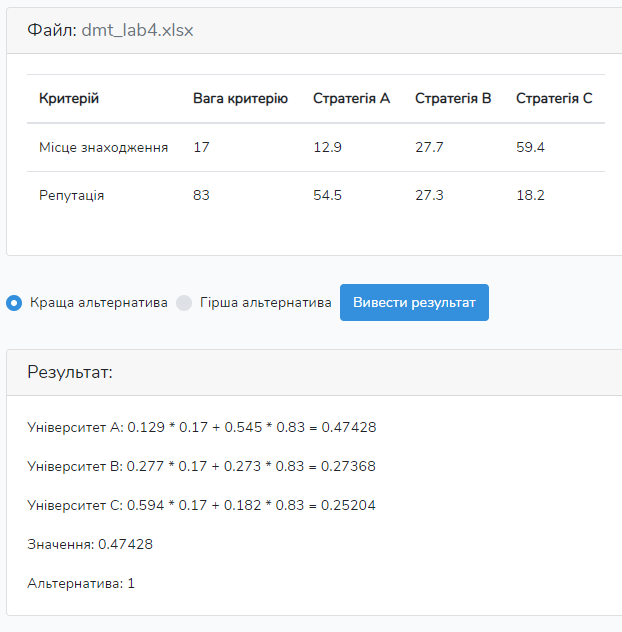
**return $result\_data;**

**}**

**//print analysis\_of\_hierarchies\_calculate($data)["val"] . PHP\_EOL;**

**}**

**Результат роботи програми:**

Результат для методу аналізу ієрархій (краща альтернатива).

**Висновок:** Під час проведення цієї лабораторної роботи я засвоїв навички використання методу аналізу ієрархій для вирішення задач системного аналізу. Дослідив роботу цього методу на даних з ексель файлу.

**ЛАБОРАТОРНА РОБОТА №5**

**Тема:** Використання теорії ігор при прийняття рішень на основі груп конфліктних критеріїв.

**Мета:** Зaкріпити навички використання теорії ігор при оптимізації елементної бази засобів вимірювання (ЗВ).

**Теоретичні відомості:** Задачі з неповною інформацією вирішуються методами пасивних ігор. В пасивній грі один гравець є активним (проектувальник), який намагається створити ЗВ, що відповідають ТЗ. Іншим гравцем є невідомі невизначені умови, що узагальнено звуться природою. Це створює конфліктну ситуацію, якій приймають участь дві сторони або два гравця з разною метою.

Модель конфліктної ситуації і правила її вирішення звуться «грою».

Кожний гравець має певний набор дій, що направлені на досягнення мети. Ці дії називаються стратегіями.

Конкретна дія, яку виконує гравець із набору можливих, зветься ходом.

В процесі функціонального проектування ЗВ вирішуються дві

задачі:1) вибір структури; 2) вибір та обгрунтування її конструктивних модулів (елементної бази).

Рішення про вибір елементної бази ЗВ приймаються з врахуванням конструкторсько-технологічних обмежень. Складності, що виникають при виборі та обгрунтуванні елементної бази, пов’язані з необхідністю узгодження елементів по інформативним, енергетичним, конструктивним і технологічним ознакам.

Багато елементів функціональної або електричної принципової схеми ЗВ може бути «покрито» різними конструктивними модулями, що забезпечують функціонування ЗВ.

Серед показників якості ЗВ містяться дві конфліктуючі групи: вартість та надійність, вартість і точність, габаритні розміри і міцність і т. д. Це означає, що одні показники необхідно зменшувати, а інші збільшувати. Для рішення цієї задачі доцільно використати теорію ігор.

Прийнявши за глобальний критерій ефективність, складемо дві ігрові матриці. В кожну з них занесемо неконфліктуючі між собою показники якості ЗВ. В строчках матриць запишемо можливі варіанти «покриття» х і , а в стовбчиках - показники ЗВ (к j ). В комірці ігрової матриці запишемо ефективності  i,j .

Обидві матриці мають однакову кількість строчок, що відображають можливі стратегії «покриття» схеми ЗВ конструктивними модулями. Кількість стовбчиків в ігрових матрицях може бути різним в залежності від кількості показників якості ЗВ.

В одній ігровій матриці показники ефективності ЗВ максимізуються, в іншій - мінімізуються.

Показник ефективності ЗВ по критерію надійності визначається співвідношенням:

Е i,j (p)=P i,j /c i , (1)

де P i,j - ймовірність безвідмовної роботи сукупності модулів, що необхідні для реалізації схеми по і-й стратегії;

с і - вартість модулів, що використовуються при і-й стратегії.

Показник ефективності ЗВ по швидкодії орієнтовно модна знайти, використавши співвідношення:

Е i,j (t)=1/c i ,t i (2)

де t i - паспортні дані про час виконання типової обчислювальної операції елементною базою, вибраною по і-й стратегії.

Показник ефективості ЗВ по критерію мінімального енергоспоживання у випадку мінімізуємої матриці вираховується по формулі:

Е i,j (w)=1/W i c, (3)

де W i - енергія, що споживається всіма модулями при виборі елементної бази по і-й стратгії;

с - коефіцієнт вартості одиниці енергії.

Показник ефективості ЗВ по кількості зовнішніх зв’зків модулів, що застосовуються по і-й стратегії, можуть характеризувати трудомісткість і надійність ЗВ. В задачах мінімізації цей показник визначається виразом:

Е i,j (n)=1/N i , (4)

де N i - загальна кільеість зовнішніх виводів конструктивних модулів, що застосовуються у виробі по і-й стратегії.

**Код програми:**

**<?php**

**namespace App\DMT;**

**class Lab5 {**

**public $dmt\_spreadsheet = array();**

**public function \_\_construct($data)**

**{**

**$this->dmt\_spreadsheet = $data;**

**}**

**public function theory\_of\_game\_calculate($data, $coefficient, $maximize\_params, $best\_alternative = true) {**

**$result\_data = array();**

**$efficiency\_matrix = array();**

**$normalize\_matrix = array();**

**$weight\_coefficient\_matrix = array();**

**$values\_by\_params = array();**

**$min\_or\_max\_values\_by\_params = array();**

**$sum\_of\_minimize\_params = array();**

**$sum\_of\_maximize\_params = array();**

**$efficiency\_matrix = $data;**

**foreach ($efficiency\_matrix as $key => $row) {**

**$Ec = $row[count($row) - 1];**

**foreach ($row as $index => $column) {**

**if ($index != 0) {**

**switch ($index) {**

**case 1:**

**$efficiency\_matrix[$key][$index] = 1 / ($efficiency\_matrix[$key][$index] \* $Ec);**

**break;**

**case 2:**

**$efficiency\_matrix[$key][$index] = $efficiency\_matrix[$key][$index] / $Ec;**

**break;**

**case 3:**

**$efficiency\_matrix[$key][$index] = 1 / ($efficiency\_matrix[$key][$index] \* $Ec);**

**break;**

**case 4:**

**$efficiency\_matrix[$key][$index] = $efficiency\_matrix[$key][$index] / $Ec;**

**break;**

**case 5:**

**break;**

**}**

**}**

**}**

**}**

**foreach ($efficiency\_matrix as $key => $row) {**

**if ($key == 0) {**

**for ($i = 0; $i < count($row) - 1; $i++) {**

**array\_push($values\_by\_params, array());**

**}**

**}**

**foreach ($row as $index => $column) {**

**if ($index != 0) {**

**array\_push($values\_by\_params[$index - 1], $column);**

**}**

**}**

**}**

**foreach ($values\_by\_params as $key => $row) {**

**$min\_or\_max\_values = $row[0];**

**foreach ($row as $index => $column) {**

**if ($maximize\_params[$key]) {**

**if ($column > $min\_or\_max\_values) $min\_or\_max\_values = $column;**

**} else {**

**if ($column < $min\_or\_max\_values) $min\_or\_max\_values = $column;**

**}**

**}**

**array\_push($min\_or\_max\_values\_by\_params, $min\_or\_max\_values);**

**}**

**$normalize\_matrix = $efficiency\_matrix;**

**foreach ($normalize\_matrix as $key => $row) {**

**foreach ($row as $index => $column) {**

**if ($index != 0) {**

**if ($maximize\_params[$key]) {**

**$normalize\_matrix[$key][$index] /= $min\_or\_max\_values\_by\_params[$index - 1];**

**} else {**

**$normalize\_matrix[$key][$index] = $min\_or\_max\_values\_by\_params[$index - 1] / $efficiency\_matrix[$key][$index];**

**}**

**}**

**}**

**}**

**$weight\_coefficient\_matrix = $normalize\_matrix;**

**foreach ($weight\_coefficient\_matrix as $key => $row) {**

**foreach ($row as $index => $column) {**

**if ($index != 0) {**

**$weight\_coefficient\_matrix[$key][$index] = $weight\_coefficient\_matrix[$key][$index] \* $coefficient[$index - 1];**

**}**

**}**

**}**

**foreach ($weight\_coefficient\_matrix as $key => $row) {**

**$sum\_of\_max = 0;**

**$sum\_of\_min = 0;**

**foreach ($row as $index => $column) {**

**if ($index != 0) {**

**if ($maximize\_params[$index - 1]) {**

**$sum\_of\_max += $column;**

**} else {**

**$sum\_of\_min += $column;**

**}**

**}**

**}**

**array\_push($sum\_of\_maximize\_params, $sum\_of\_max);**

**array\_push($sum\_of\_minimize\_params, $sum\_of\_min);**

**}**

**$val = $sum\_of\_maximize\_params[0] / $sum\_of\_minimize\_params[0];**

**$alternative = 0;**

**for ($i = 0; $i < count($weight\_coefficient\_matrix); $i++) {**

**if ($best\_alternative) {**

**if ($sum\_of\_maximize\_params[$i] / $sum\_of\_minimize\_params[$i] > $val) {**

**$val = $sum\_of\_maximize\_params[$i] / $sum\_of\_minimize\_params[$i];**

**$alternative = $i;**

**}**

**} else {**

**if ($sum\_of\_maximize\_params[$i] / $sum\_of\_minimize\_params[$i] < $val) {**

**$val = $sum\_of\_maximize\_params[$i] / $sum\_of\_minimize\_params[$i];**

**$alternative = $i;**

**}**

**}**

**}**

**$result\_data["val"] = $val;**

**$result\_data["alternative"] = $alternative;**

**$result\_data["efficiency\_matrix"] = $efficiency\_matrix;**

**$result\_data["normalize\_matrix"] = $normalize\_matrix;**

**$result\_data["weight\_coefficient\_matrix"] = $weight\_coefficient\_matrix;**

**$result\_data["min\_or\_max\_values\_by\_params"] = $min\_or\_max\_values\_by\_params;**

**$result\_data["sum\_of\_minimize\_params"] = $sum\_of\_minimize\_params;**

**$result\_data["sum\_of\_maximize\_params"] = $sum\_of\_maximize\_params;**

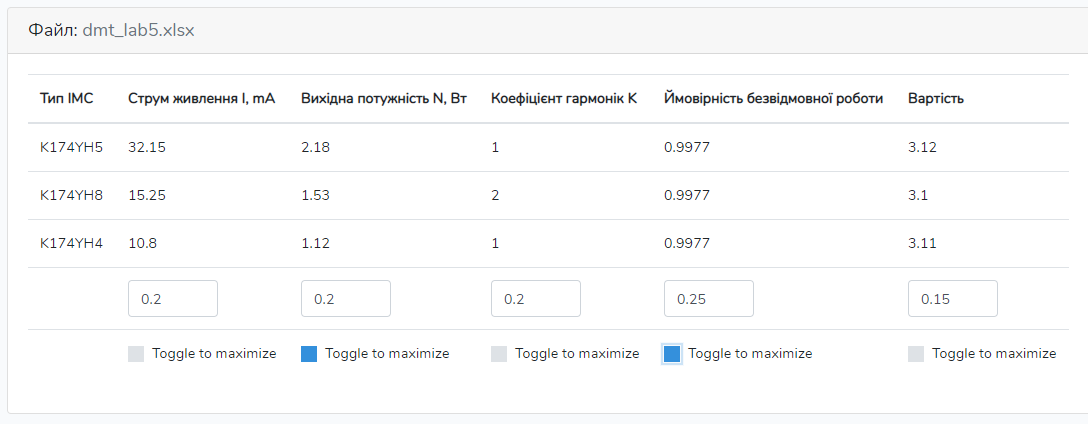
**return $result\_data;**

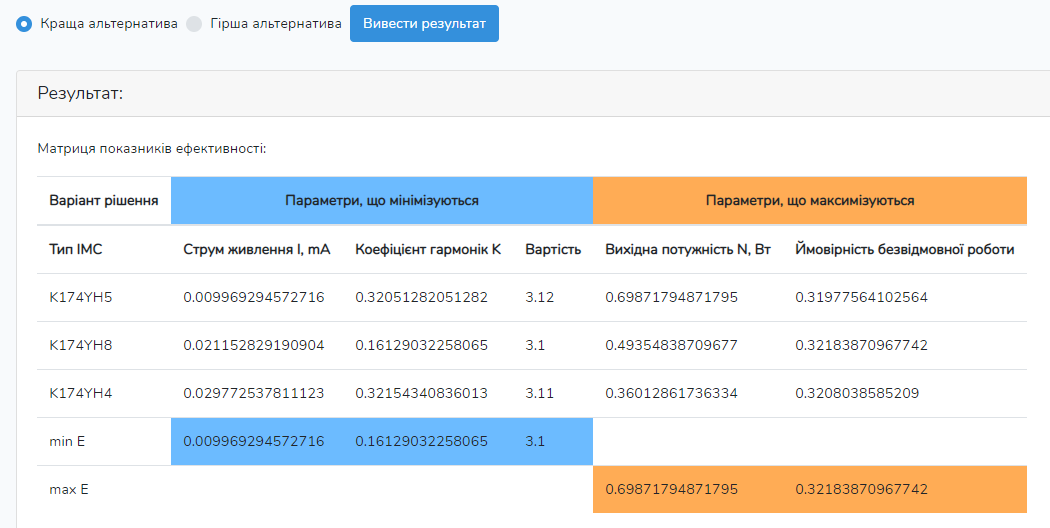
**}**

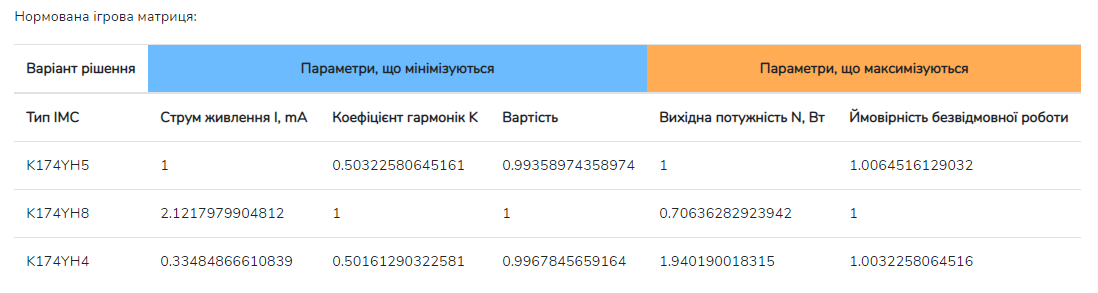
**//print theory\_of\_game\_calculate($data, array(0.2, 0.2, 0.2, 0.25, 0.15), array(false, true, false, true, false))["val"] . PHP\_EOL;**

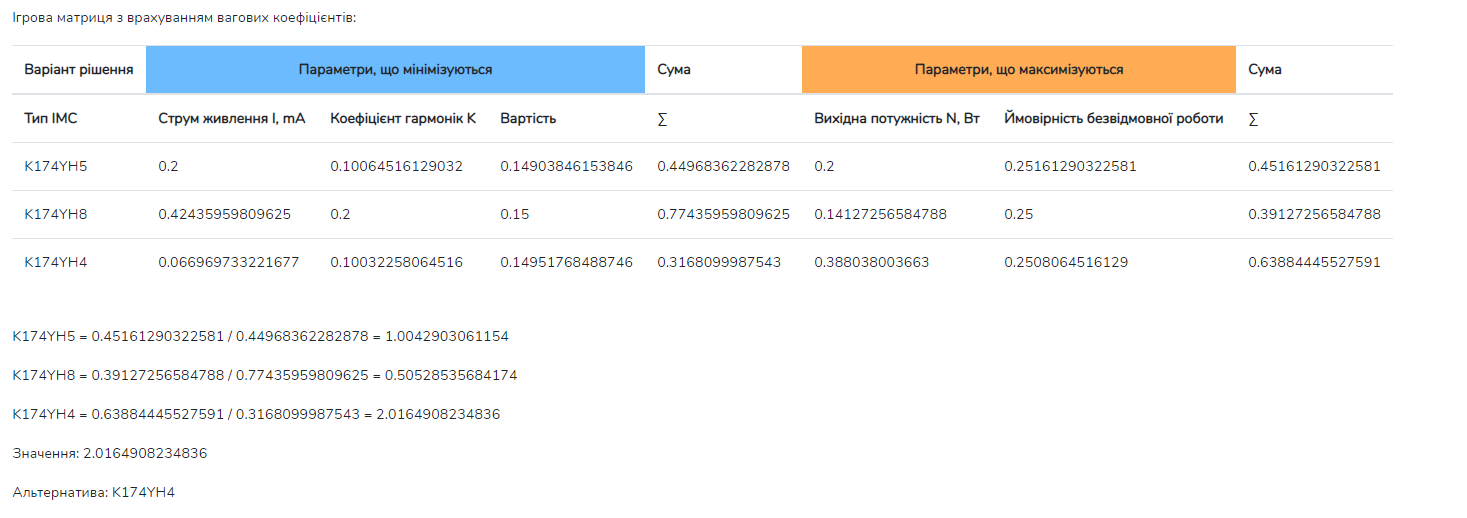
**}**

**Результат роботи програми:**

Після того як ми загрузили ексель файл, появилася таблиця з данимии. Під кожним параметром є поле для введення коефіцієнту цього параметру. А під цим полем є чекбокс, який відповідає за максимізацію цього параметру.

Матриця показників ефективності.

Нормована ігрова матриця.

Ігрова матриця з врахуванням вагових коефіцієнтів. Під матрицею виведений результат роботи методу

**Висновок:** Під час проведення цієї лабораторної роботи я засвоїв навички використання теорії ігор при прийнятті рішень на основі груп конфліктних критеріїв. Дослідив роботу цього методу на даних з ексель файлу.

**ЛАБОРАТОРНА РОБОТА №6**

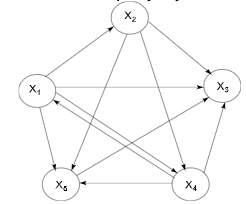
**Тема:** Метод розстановки приорітетів.

**Мета:** Закріпити навички використання методу розстановки приорітетів при вирішенні багатоцільових задач прийняття рішень.

**Теоретичні відомості:** При вирішенні задач багатоцільової оптимізації часто приходиться потрівнювати альтернативні варіанти по ступеню їх важливості, або, як говорять, проводити ранжування альтернатив. Часто в цьому випадку використовується метод експертних оцінок.

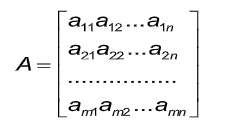
В початковому вигляді задача розстановки приорітетів відома як «задача про лідера», в якій розглядається проблема визначення результатів деякого спортивного турніра. Той порядок визначення переможця (лідера) і розподіл місць сред іних учасників туріра, який використовується на даний час і суть якого в отриманні суми балів кожного гравця або команди, не завжди може бути визнаним безпомилковим. В цьму випадку місце гравця в турнірній таблиці визначає суму балів, що отримана без врахування сили суперників, у яких виграв даний гравець.

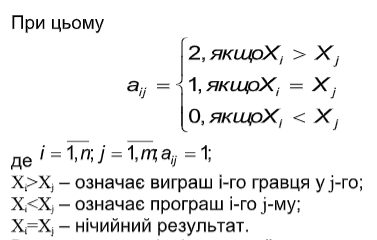
Розглянемо інший підход до рішення «задачі про лідера». Подамо результати турніру n гравців у вигляді деякого орієнтовного графа. Кожному із n учасників (x 1 , x 2 , ..., x n ) відповідає вершина графа. Якщо гравець х і виграв у гравця х j , то на графі є дуга ij. Нічийному результату відповідають дуги в прямому і зворотньому напрямку: ij та ji. Приклад такого графа поданий на рисунку 1.



Метод вирішення задачі реалізується таким чином. Будується

матриця Aaij





Вводиться термін ітерованої «сили» порядка k гравця Х і . Ітерована сила першого порядку гравця Х і позначається Р і (1) і знаходиться у вигляді суми балів данного гравця. При цьому не враховується «сила» суперників.

Розподіл балів серед гравців задаєься вектором:

Р(1)=[P 1 (1), P 2 (1), ..., P n (1)].

На другій ітерації за «силу» гравця прийається ітерована сила першого порядка.

Ітерована «сила» другого порядку розраховується з врахуванням «сил» суперників.

В загальному вигляді вона подається вектором:

P(2)= [P 1 (2), P 2 (2), ..., P n (2)].

Подальші розрахунки проводяться аналогічно:

P(k)=AP(k–1),

при цьому

Р(0)=(1, 1, ..., 1).

Процес розрахунку являє собою послідовному застосуванні перетворення, що задається матрицею А, до начального векору Р(0). Позначимо через Р і 0 (k) нормовану ітеровану силу k-го порядка і- го гравця.

**Код програми:**

**<?php**

**namespace App\DMT;**

**class Lab6 {**

**public $dmt\_spreadsheet = array();**

**public function \_\_construct($data)**

**{**

**$this->dmt\_spreadsheet = $data;**

**}**

**public function priority\_setting\_calculate($data) {**

**$result\_data = array();**

**$adjacency\_matrix = array();**

**$sum\_by\_row = array();**

**$p = array();**

**foreach ($data as $index => $column) {**

**$adjacency\_matrix[$index] = array();**

**foreach ($data as $elem) {**

**if ($column > $elem) {**

**array\_push($adjacency\_matrix[$index], 2);**

**} else if ($column < $elem) {**

**array\_push($adjacency\_matrix[$index], 0);**

**} else if ($column == $elem) {**

**array\_push($adjacency\_matrix[$index], 1);**

**}**

**}**

**}**

**foreach ($adjacency\_matrix as $row) {**

**array\_push($sum\_by\_row, array\_sum($row));**

**}**

**$p[0] = array();**

**$sum = array\_sum($sum\_by\_row);**

**foreach ($sum\_by\_row as $row) {**

**array\_push($p[0], $row / $sum);**

**}**

**$p[1] = array();**

**foreach ($adjacency\_matrix as $row) {**

**$s = 0;**

**foreach ($row as $index => $column) {**

**$s += $column \* $sum\_by\_row[$index];**

**}**

**array\_push($p[1], $s);**

**}**

**$p[2] = array();**

**$sum2 = array\_sum($p[1]);**

**foreach ($p[1] as $row) {**

**array\_push($p[2], $row / $sum2);**

**}**

**$result\_data["adjacency\_matrix"] = $adjacency\_matrix;**

**$result\_data["sum\_by\_row"] = $sum\_by\_row;**

**$result\_data["p"] = $p;**

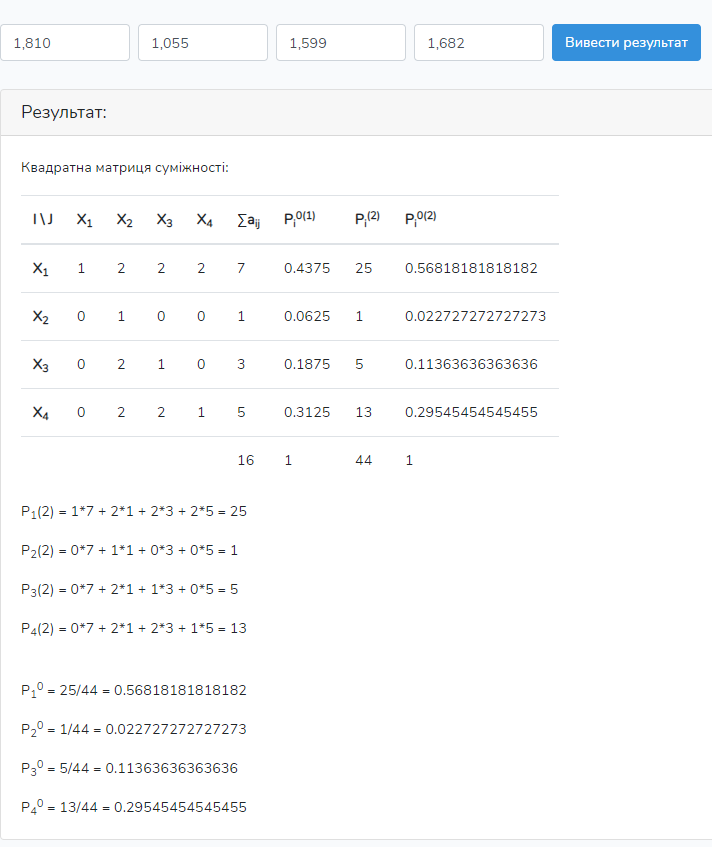
**return $result\_data;**

**}**

**//print priority\_setting\_calculate($data) . PHP\_EOL;**

**}**

**Результат роботи програми:**

Спочатку потрібно ввести значення X1..4. Після натискання кнопки “Вивести результат” виводиться квадратна матриця суміжності. Під цією матрицею виводиться покрокове обчислення значень.

**Висновок:** Під час проведення цієї лабораторної роботи я засвоїв навички використання методу розтановки приорітетів при вирішенні багатоцільових задач прийняття рішень.

**ЛАБОРАТОРНА РОБОТА №7**

**Тема:** Графічний метод вирішення задач оптимізації.

**Мета:** Закріпити навички використання графічного методу при вирішенні оптимізаційних задач.

**Теоретичні відомості:** Під оптимізацією прийняття рішення розуміють процес вибору із кількох можливих варіантів того рішення, яке відповідає висунутим вимогам (критеріям). Оптимізаційні задачі частіше всього розв’язуються математичними методами. Їх можна застосовувати тільки до математичних моделей того чи іншого явища. Тому особливо важливу роль має переведення звичайної умови задачі в її математичну модель.

Математична модель. Умову задачі можливо записати за допомогою математичних рівнянь чи нерівностей. Це і є математична модель дано задачі. Використовуються такі методи отримання математичних моделей:

1) теоретико-аналітичний;

2) експериментально-статистичний;

3) статистичного моделювання (Монте-Карло).

Цільова функція. Основою математичної моделі є деяка функція, що здійснює переклад запитання задачі із звичайної мови на мову функцій. Цей вираз є критерієм якості, що дозволяє кількісно порівняти два альтернативних рішення. Ця функція зветься цільовою (ЦФ). Значення цього виразу інженер намагається зробити максимальним або мінімальним. Отриманий при цьому результат і дає розв’язок задачі оптимізації. Із математичної точки зору цільова функція описує деяку (n+1)-мірну поверхню. Її характеристики визначаються проектними параметрами.

M=m(х 1 , х 2 , ..., х n ).

В інженерній практиці прикладами цільової функції часто є вартість, міцність, габаритні розміри, ККД.

Проектні параметри. Це незалежні змінні параметри, які повністю і однозначно визначають задачу проектування, що вирішується. Проектні параметри - це невідомі веичини, значення яких обчислюються в процесі оптимізації. Проектними параметрами можуть бути будьякі основні або похідні величини, що служать для кількістного опису системи. Це можуть бути невідомі значення довжини, маси, часу, температури. Кількість проектних параметрів характеризує суінь складності задачі проектування. Кількість проектрих параметрів позначають “n”, а самі проектні параметри - “x” з відповідними індексами. Таким чином n проектних параметрів будемо позначати таким чином: х 1 , х 2 , ... ,х n .

Простір проектування. Простір проектування - область, що визначена всіма n проектними параметрами (обмеженнями). Розрізняють обеження- рівняння і обмеження-нерівності.

Обмеження-рівняння. Це залежність між проектними параметами, котрі повинні враховуватись при відшуканні рішення. Вони відображають закони природи, економіки, наявність необхідних матеріалів і т. п.. Кількість обмежень-рівнянь може бути любою.

C 1 (x 1 , x 2 , ..., x n )=0;

C 2 (x 1 , x 2 , ..., x n )=0;

C n (x 1 , x 2 , ..., x n )=0.

Якщо яке-небудь із цих співвідношень можливо вирішити відносно одного із проектних параметрів, то можливо виключити цей параметр із процесу оптимізації, тим самим зменшуючи число вимірності простору проектування і спрощуєтючи рішення задачі.

Обмеження-нерівності. Це обмеження, виражені нерівностями. В загальному випадку їх може бути скільки завгодно.

Часто, в зв’язку з обмеженнями, оптимальне значення цільової функції буває на одній з границь області проектування, а не там, де її поверхня має нульовий градієнт.

Складність рішення оптимізаційної задачі залежить від:

1) вигляду функції мети;

2) кількості проектних параметрів;

3) вигляду та кількості обмежень.

Задачі лінійного програмування самі прості. Для них функція мети лінійна, обмеження мають вигляд лінійних рівностей або нерівностей.

Ідея графічного метода вирішення оптимізаційних задач, що мають два проектних параметра була запропонована академіком Л. В. Канторовичем в 1939 році.

Цільва функція та обеження задачі визначаються у вигляді графіків, осі координат яких задаються проектними параметрами. Простір проектування визначається графіками обмежень. Переміщаючи графік функції мети паралельно в сторону збільшення або зменшення її значення (відповідно при завданні максимізації або мінімізації ЦФ) знаходимо найбільшу (найменшу) спільну точку функції мети і проектного простору. Координати цієї точки і визначають оптимальні значення проектних параметрів.

**Особисте завдання:** Завод по випуску комплектуючих виробляє два типа транзисторів для зборки радіорелейних станцій: типу А і типу Б. Норми використання матеріалів при виробництві транзисторів, максимальний об’єм використаних матеріалів та прибуток від реалізації 10 000 транзисторів подані в таблиці. Необхідно скласти план виробництва транзисторів типу А і Б з метою максимізації сумарного прибутку.





**Хід виконання роботи:**

**Цільова функція:** F = 572x1+290x2 → max, при системі обмежень:

0.23x1+0.2x2≤600,

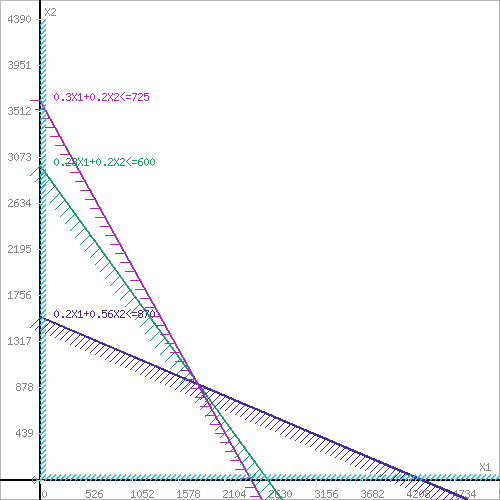
0.2x1+0.56x2≤870,

0.3x1+0.2x2≤725,

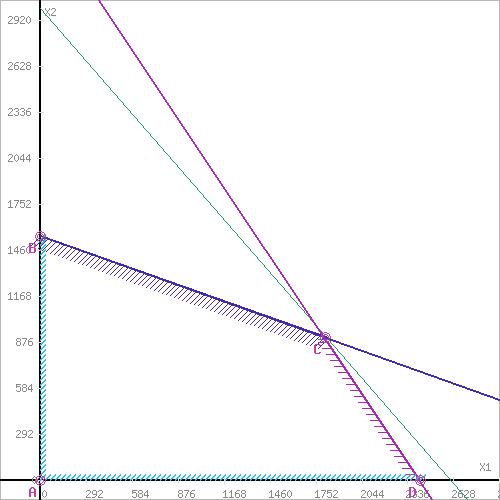
x1 ≥ 0,

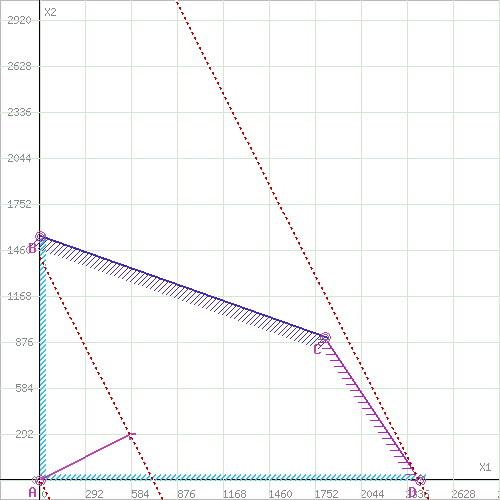
x2 ≥ 0.

Спочатку потрібно побудувати область допустимих значень. Для цього побудуємо лінії які відповідають функціям системи обмежень.



Перетином напівплощин буде область, координати точок якого задовольняють умові нерівностей системи обмежень задачі.

Побудуємо пряму, що відповідає значенню функції F = 572x1 + 290x2 = 0. Вектор-градієнт, складений з коефіцієнтів цільової функції, вказує напрямок максимізації F (X). Початок вектора - точка (0; 0), кінець - точка (572; 290). Будемо рухати цю пряму паралельним чином. Оскільки нас цікавить максимальне рішення, тому рухаємо пряму до останнього торкання позначеної області.

Пряма F(x) перетинає область в точці D. Її координати задовольняють рівнянням цих прямих:

x2 = 0,

0.3x1 + 0.2x2 = 725.

Вирішивши систему рівнянь, отримаємо: x1 = 2416, x2 = 0.

Максимальне значення цільової функції: F (X) = 572 \* 2416 + 290 \* 0 = ‭1381952‬.

Максимальний прибуток становить: 138,19 грн.

**Висновок:** Під час проведення цієї лабораторної роботи я засвоїв графічний метод вирішення оптимізаційних задач.