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:  Професор каф. ПЗАС  Голуб С.В.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2020 р. | Виконав:  студент 1-го курсу  групи МПЗ-1904  Гаврилюк В. Є. |

Черкаси 2020

# **Лабораторна робота №1**

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

**Мета:** 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, 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 (HA), 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 рj. Значення кожного рядка підсумовуються. До сум застосовується max і номер того рядка, в якому є найбільша сума (Z BL ), дає кращу альтернативу.

**Код програми:**

<?

namespace App\TPR;

class Lab1 {

public $tprSpreadsheet = array();

public function \_\_construct($data) {

$this->tprSpreadsheet = $data;

}

public function maxymaxCalculate($data, $bestAlternative = true) {

$value = $data[0][0];

$numberRowAlternative = 0;

$resultData = array();

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

foreach ($row as $column) {

if ($bestAlternative) {

if ($column > $value) {

$value = $column;

$numberRowAlternative = $index;

}

} else {

if ($column < $value) {

$value = $column;

$numberRowAlternative = $index;

}

}

}

}

$resultData["value"] = $value;

$resultData["numberRowAlternative"] = $numberRowAlternative;

return $resultData;

}

public function valdaCalculate($data, $bestAlternative = true) {

$value = -1;

$numberRowAlternative = 0;

$resultData = array();

$valuesFromEachRow = array();

foreach ($data as $row) {

$suitableValueFromRow = $row[0];

foreach ($row as $column) {

if ($bestAlternative) {

if ($column < $suitableValueFromRow)

$suitableValueFromRow = $column;

} else {

if ($column > $suitableValueFromRow)

$suitableValueFromRow = $column;

}

}

array\_push($valuesFromEachRow, $suitableValueFromRow);

}

$value = $valuesFromEachRow[0];

foreach ($valuesFromEachRow as $index => $suitableValue) {

if ($bestAlternative) {

if ($suitableValue > $value) {

$value = $suitableValue;

$numberRowAlternative = $index;

}

} else {

if ($suitableValue < $value) {

$value = $suitableValue;

$numberRowAlternative = $index;

}

}

}

$resultData["value"] = $value;

$resultData["numberRowAlternative"] = $numberRowAlternative;

return $resultData;

}

public function gurvitsaCalculate($data, $q, $bestAlternative = true) {

$value = -1;

$numberRowAlternative = 0;

$resultData = array();

$minValuesFromEachRow = array();

$maxValuesFromEachRow = array();

foreach ($data as $row) {

$suitableMinValueFromRow = $row[0];

$suitableMaxValueFromRow = $row[0];

foreach ($row as $column) {

if ($bestAlternative) {

if ($column < $suitableMinValueFromRow)

$suitableMinValueFromRow = $column;

if ($column > $suitableMaxValueFromRow)

$suitableMaxValueFromRow = $column;

} else {

if ($column > $suitableMinValueFromRow)

$suitableMinValueFromRow = $column;

if ($column < $suitableMaxValueFromRow)

$suitableMaxValueFromRow = $column;

}

}

array\_push($minValuesFromEachRow, $suitableMinValueFromRow);

array\_push($maxValuesFromEachRow, $suitableMaxValueFromRow);

}

$value = $q \* $minValuesFromEachRow[0] + (1 - $q) \* $maxValuesFromEachRow[0];

for ($i = 0; $i < count($minValuesFromEachRow); $i++) {

$calculatedValues = $q \* $minValuesFromEachRow[$i] + (1 - $q) \* $maxValuesFromEachRow[$i];

if ($bestAlternative) {

if ($calculatedValues > $value) {

$value = $calculatedValues;

$numberRowAlternative = $i;

}

} else {

if ($calculatedValues < $value) {

$value = $calculatedValues;

$numberRowAlternative = $i;

}

}

}

$resultData["value"] = $value;

$resultData["numberRowAlternative"] = $numberRowAlternative;

return $resultData;

}

public function bayesaLaplasaCalculate($data, $probability, $bestAlternative = true) {

$value = -1;

$numberRowAlternative = 0;

$resultData = array();

$newDataArray = array();

foreach ($data as $row) {

$newRowData = array();

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

array\_push($newRowData, $column \* $probability[$index]);

}

array\_push($newDataArray, $newRowData);

}

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

foreach ($newDataArray as $index => $row) {

if ($bestAlternative) {

if (array\_sum($row) > $value) {

$value = array\_sum($row);

$numberRowAlternative = $index;

}

} else {

if (array\_sum($row) < $value) {

$value = array\_sum($row);

$numberRowAlternative = $index;

}

}

}

$resultData["value"] = $value;

$resultData["numberRowAlternative"] = $numberRowAlternative;

return $resultData;

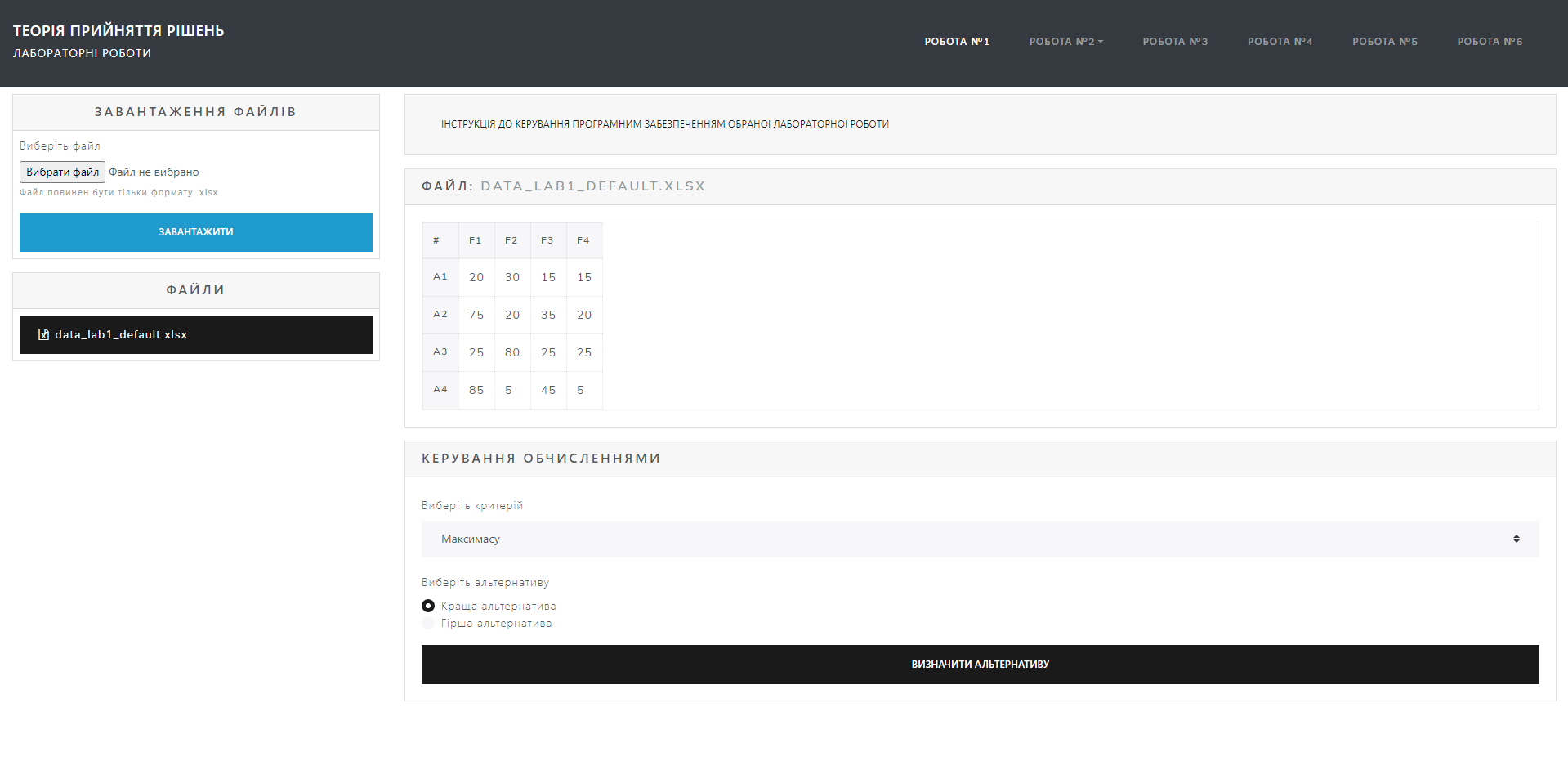
}

}

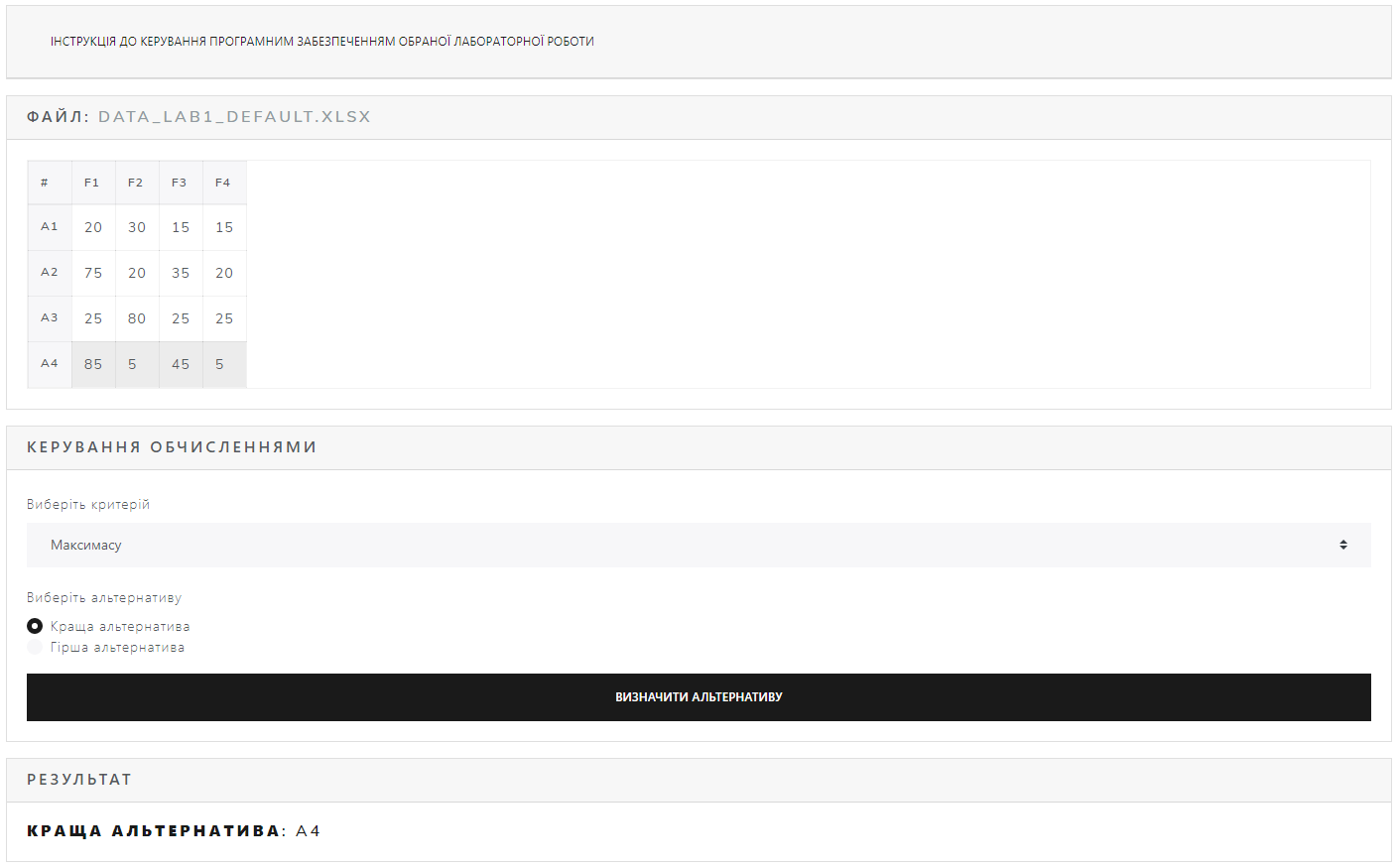
?>

**Результат роботи програми:**

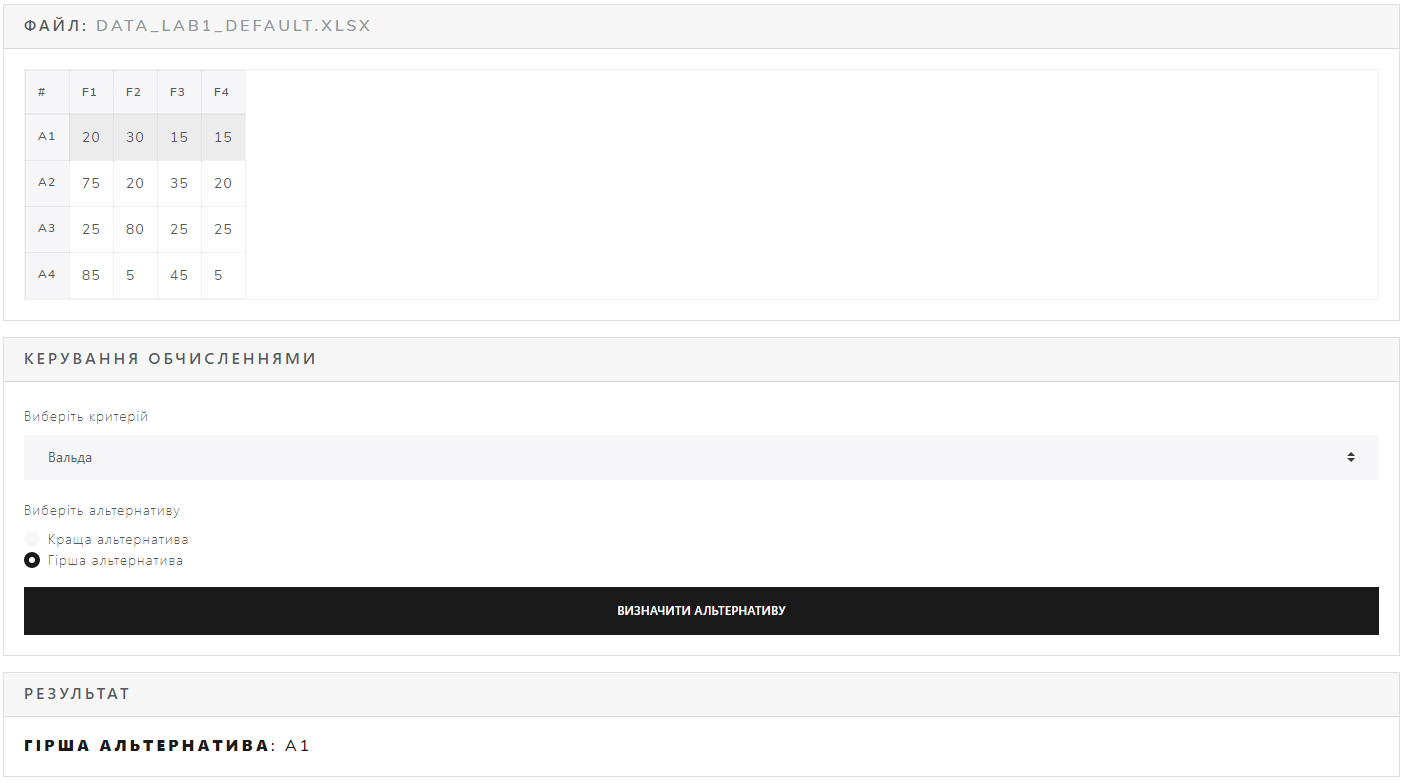
Для початку потрібно завантажити excel-файл із вхідними даними. Для цього потрібно скористатися формою «Завантаження файлів», що знаходиться зліва на робочому просторі програми. За допомогою вікна вибору обрати файл на комп’ютері та натиснути кнопку «Завантажити». Після завантаження, файл стане доступним у панелі «Файли», яка знаходиться в лівій частині робочого простору програми під формою завантаження файлів. Також, відразу після завантаження, в панелі «Файл», що знаходиться в правій частині робочого простору програми під інструкцією користувача, стануть доступними для перегляду завантажені вхідні дані у вигляді таблиці. Під нею є додаткова панель «Керування обчисленнями», що дозволяє виставляти різні параметри для проведення обрахунків. Додаткові інструкції з використання даного програмного забезпечення наведені в панелі «Інструкція до керування програмним забезпеченням обраної лабораторної роботи».



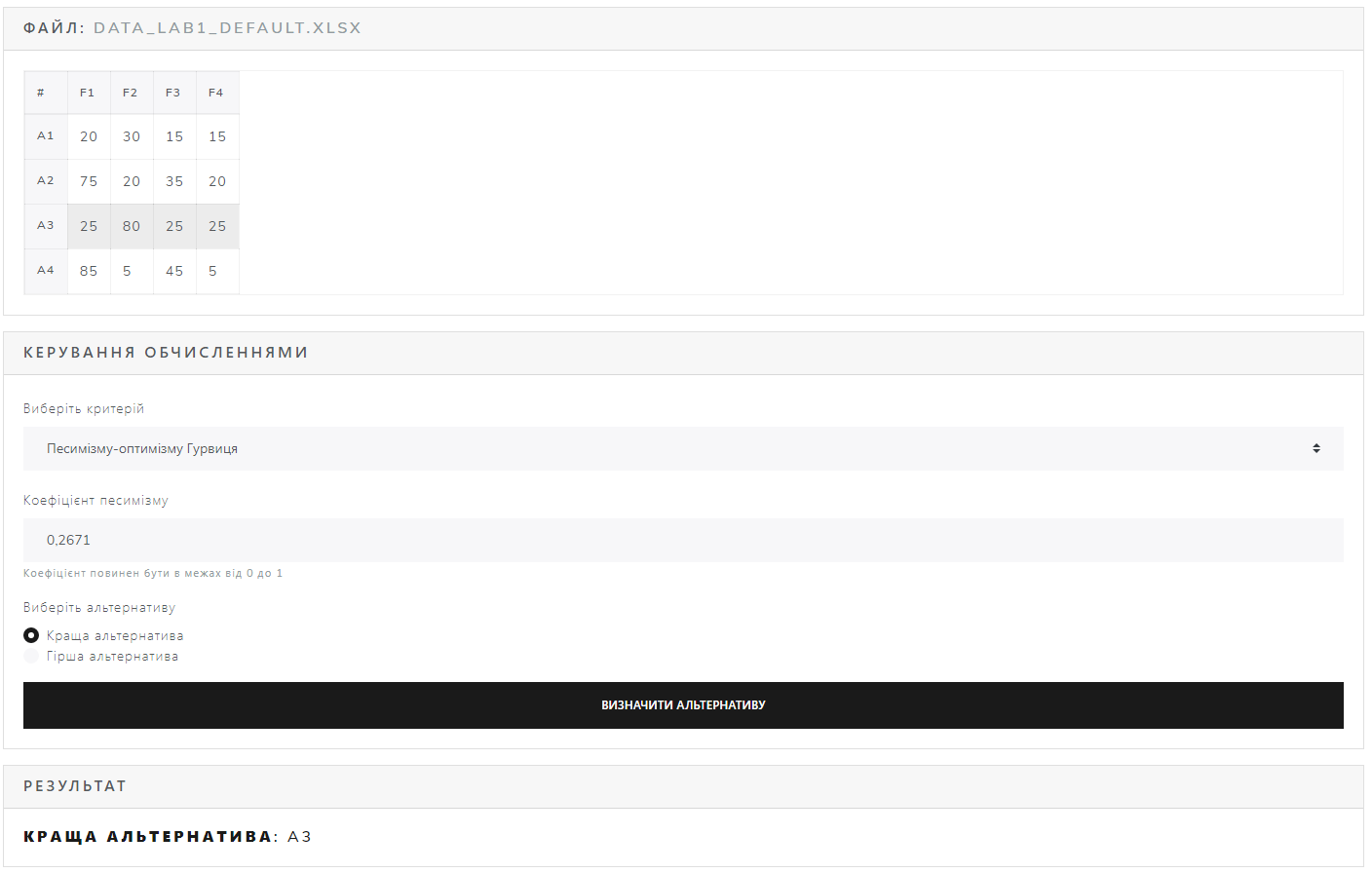
*Рис. 1.1. Вигляд робочого простору після завантаження вхідних даних*



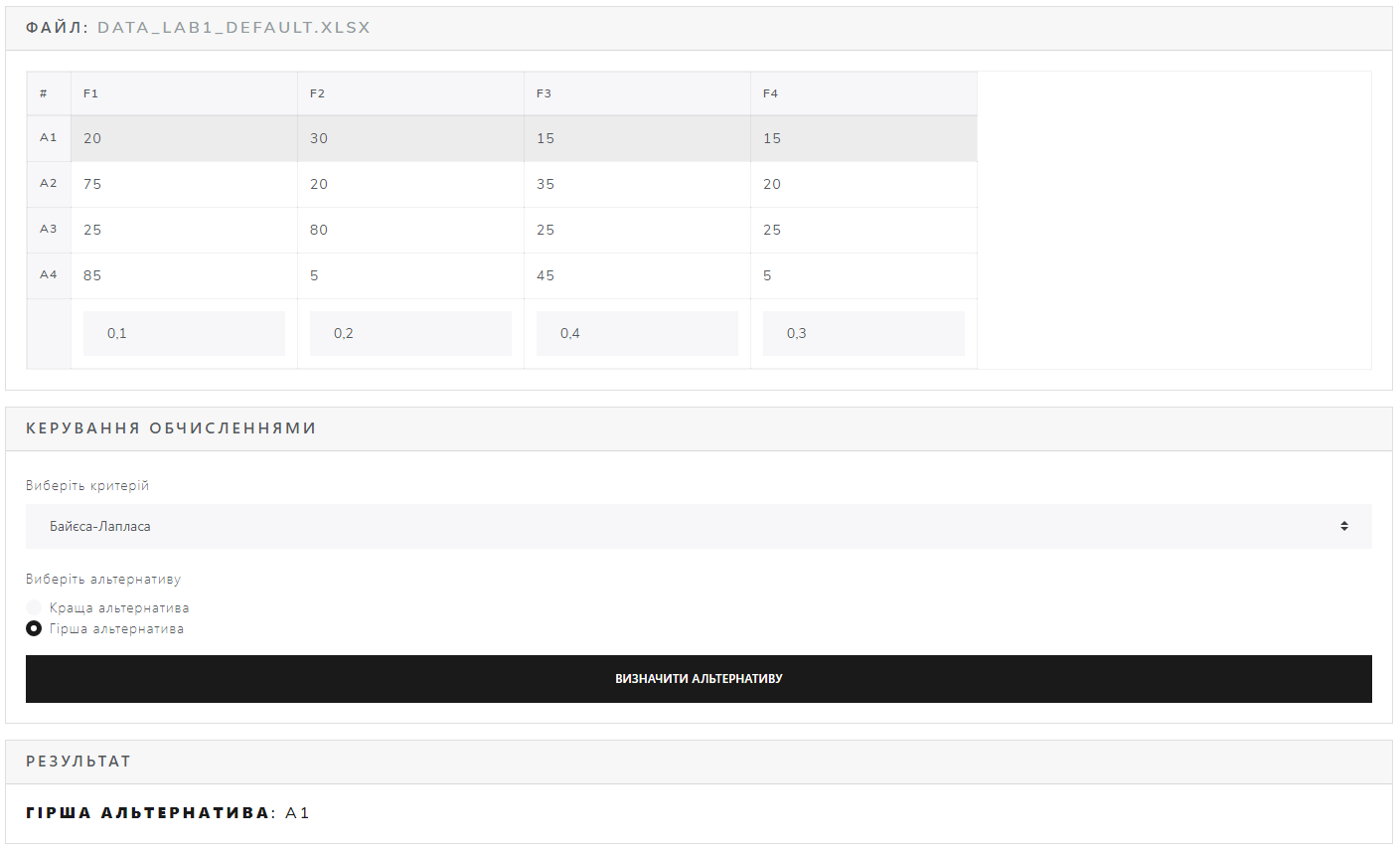
*Рис. 1.2. Визначення кращої альтернативи для критерію Максимасу*



*Рис. 1.3. Визначення гіршої альтернативи для критерію Вальда*



*Рис. 1.4. Визначення кращої альтернативи для критерію песимізму-оптимізму Гурвиця*



*Рис. 1.5. Визначення гіршої альтернативи для критерію Байєса-Лапласа*

**Висновок**: під час виконання цієї лабораторної роботи я провів аналіз альтернатив в умовах невизначеності. Засвоїв навички оптимізації рішень в умовах невизначеності. Ознайомився з критеріями визначення альтернатив: критерієм Максимасу, максимінним критерієм Вальда, критерієм песимізму-оптимізму Гурвиця та критерієм Байєса-Лапласа. Створив web-програму для визначення альтернативи в умовах невизначеності на основі вхідних даних у форматі excel-файлу.

# **Лабораторна робота №2**

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

**Мета:** Засвоєння навичок аналізу альтернатив в умовах ризику.

**Теоретичні відомості:** *Критерій* *мінімаксного ризику Севіджа*. При виборі стратегії ОПР керується не матрицею виграшів *А*, а матрицею ризиків *R*: 

*Критерій очікуваного значення*(КОЗ). Використання очікуваних величин потребує багаторазового вирішення однієї задачі, поки не будуть отримані досить точні розрахункові формули. *Середнє очікуване значення* є середньозваженим для всіх можливих результатів, де ймовірність кожного результату використовується як частота або вага відповідного значення. У загальному випадку задача ухвалення рішень може включати *n* станів природи і *m* альтернатив. Якщо *p*j – вірогідність *j*-го стану природи, а *a*ij – платіж, пов'язаний з ухваленням рішення *i* при стані природи *j* (*i* = 1, 2, ..., *m*; *j* = 1, 2, ..., *n*), тоді очікуваний платіж для вирішення *i* обчислюється у вигляді**image001,**

де за визначенням *p*1+*p2*+…+*p*n=1.

Кращою альтернативою буде та, яка відповідає **image002,** або **image003**

Комбінація *очікуваного значення і дисперсії* враховує коливання (мінливість) можливого результату для рідко повторюваних ситуацій. Дисперсія – це середнє зважене з квадратів відхилень дійсних результатів від середніх очікуваних дисперсія відображає міру розкиду даних навколо середньої величини.

Якщо *х* випадкова величина з дисперсією *D*, то середнє арифметичне має дисперсію *D*/*n*, де *n* – кількість складових в . Отже, якщо *D* зменшується ймовірність того, що близько до математичного очікування, збільшується.

Формула для розрахунку *D (*або *σ2)*: Формула дисперсии,

де *xi* – *i*-е значення показника, що аналізується; *n* – кількість значень в сукупності даних, що аналізується.

Середнє квадратичне відхилення (стандартне відхилення) також характеризує міру розсіювання значень, але на відміну від дисперсії його можна порівняти з вихідними даними, тому що має ті ж одиниці виміру. Стандартне відхилення розраховується так: Среднее квадратическое (стандартное) отклонение.

Щоб зрозуміти, наскільки великий розкид щодо самих значень, потрібен відносний показник. Такий показник *V* називається коефіцієнт варіації: Коэффициент вариации.

Як видно, це відношення стандартного відхилення до середньої величини. Даний показник вимірюється у відсотках (якщо помножити на 100%). У статистиці прийнято, що, якщо значення коефіцієнта варіації менше 33%, то сукупність вважається однорідною, якщо більше 33%, то – неоднорідною. Це вважається аксіомою.

Чим більше коефіцієнт, тим сильніше коливання. Установлено наступну *якісну оцінку різних значень коефіцієнтів варіації:*

– до 10% - слабке коливання;

– 10-25% - помірне коливання;

– понад 25% - високе коливання.

Критерій граничного рівня не дає оптимального рішення, що максимізує, наприклад, прибуток або мінімізує витрати. Цей критерій не має чітко вираженого математичного формулювання і оснований в значній мірі на інтуїції і досвіді ОПР. При цьому ОПР на підставі суб'єктивних міркувань визначає найприйнятніший спосіб дій. Критерій граничного рівня звичайно не використовується, коли немає повного уявлення про безліч можливих альтернатив. Врахування ситуації ризику при цьому може вироблятися за рахунок введення законів розподілів випадкових чинників для відомих альтернатив.

Не дивлячись на відсутність формалізації критеріями граничного рівня користуються досить часто, задаючись їх значеннями на підставі експертних або дослідних даних.

**Код програми:**

<?

namespace App\TPR;

class Lab2 {

public $tprSpreadsheet = array();

public function \_\_construct($data) {

$this->tprSpreadsheet = $data;

}

public function savage\_calculate($data, $bestAlternative = true) {

$numberRowAlternative = 0;

$resultData = array();

$riskDataArray = array();

$maxValuesFromEachColumn = $data[0];

$maxValuesFromEachRow = array();

foreach ($data as $row) {

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

if ($column > $maxValuesFromEachColumn[$index]) {

$maxValuesFromEachColumn[$index] = $column;

}

}

}

foreach ($data as $row) {

$riskRowData = array();

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

array\_push($riskRowData, $maxValuesFromEachColumn[$index] - $column);

}

array\_push($riskDataArray, $riskRowData);

}

foreach ($riskDataArray as $row) {

$suitableMaxValuesFromRow = $row[0];

foreach ($row as $column) {

if ($column > $suitableMaxValuesFromRow)

$suitableMaxValuesFromRow = $column;

}

array\_push($maxValuesFromEachRow, $suitableMaxValuesFromRow);

}

$value = $maxValuesFromEachRow[0];

foreach ($maxValuesFromEachRow as $index => $maxValue) {

if ($bestAlternative) {

if ($maxValue < $value) {

$value = $maxValue;

$numberRowAlternative = $index;

}

} else {

if ($maxValue > $value) {

$value = $maxValue;

$numberRowAlternative = $index;

}

}

}

$resultData["value"] = $value;

$resultData["numberRowAlternative"] = $numberRowAlternative;

$resultData["riskDataArray"] = $riskDataArray;

return $resultData;

}

public function expectedValueCalculate($data, $c1, $c2, $n) {

$numberRowAlternative = 0;

$resultData = array();

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

$pSum = 0;

if ($index != 0) {

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

$pSum += $data[$i][0];

}

}

array\_push($data[$index], $pSum);

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

}

$value = 0;

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

if ($index != 0) {

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

$value = $data[$index][2];

$numberRowAlternative = $index;

}

}

}

$resultData["value"] = $value;

$resultData["numberRowAlternative"] = $numberRowAlternative;

$resultData["newData"] = $data;

return $resultData;

}

public function expectedValueDispersionCalculate($data, $c1, $c2, $n) {

$numberRowAlternative = 0;

$resultData = array();

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

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

$pSum = 0;

$pPowToSum = 0;

if ($index != 0) {

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

$pSum += $data[$i][0];

}

}

if ($index != 0) {

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

$pPowToSum += $data[$i][1];

}

}

array\_push($data[$index], $pSum);

array\_push($data[$index], $pPowToSum);

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));

}

$value = $data[0][4];

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

if ($row[4] < $value) {

$value = $row[4];

$numberRowAlternative = $index;

}

}

$resultData["value"] = $value;

$resultData["numberRowAlternative"] = $numberRowAlternative;

$resultData["newData"] = $data;

return $resultData;

}

public function averageExpectedValueCalculate($data, $bestAlternative = true) {

$numberRowAlternative = 0;

$resultData = array();

$casesCountArray = array();

$averageExpectedValueArray = array();

foreach ($data as $row) {

$casesCount = 0;

foreach ($row as $column) {

$casesCount += $column[1];

}

array\_push($casesCountArray, $casesCount);

}

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

$averageExpectedValue = 0;

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

$averageExpectedValue += $column[0] \* $column[1] / $casesCountArray[$index];

$data[$index][$index2][2] = $column[1] / $casesCountArray[$index];

}

array\_push($averageExpectedValueArray, $averageExpectedValue);

}

$value = $averageExpectedValueArray[0];

foreach ($averageExpectedValueArray as $index => $expectedValue) {

if ($bestAlternative) {

if ($expectedValue > $value) {

$value = $expectedValue;

$numberRowAlternative = $index;

}

} else {

if ($expectedValue < $value) {

$value = $expectedValue;

$numberRowAlternative = $index;

}

}

}

$resultData["value"] = $value;

$resultData["numberRowAlternative"] = $numberRowAlternative;

$resultData["averageExpectedValueArray"] = $averageExpectedValueArray;

$resultData["newData"] = $data;

return $resultData;

}

public function limitLevelCalculate($i1, $i2, $a1, $a2) {

$resultData = array();

$expectedDeficit = array();

$expectedSurpluses = array();

$expectedDeficitLimit = round(log($i2) - ($a1 / $i2) - 1, 3);

$expectedSurplusesLimit = round(log($i1) - ($a2 / $i2) - 1, 3);

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

array\_push($expectedDeficit, round(log($i) - ($i / $i2), 2));

array\_push($expectedSurpluses, round(log($i) - ($i / $i1), 2));

}

for ($i = 0; $i < count($expectedDeficit); $i++) {

if ($expectedDeficit[$i] >= $expectedDeficitLimit && $expectedSurpluses[$i] >= $expectedSurplusesLimit) {

$suitableStartIntervalIndex = $i;

break;

}

}

for ($i = count($expectedDeficit) - 1; $i >= 0; $i--) {

if ($expectedDeficit[$i] >= $expectedDeficitLimit && $expectedSurpluses[$i] >= $expectedSurplusesLimit) {

$suitableEndIntervalIndex = $i;

break;

}

}

$resultData["expectedDeficit"] = $expectedDeficit;

$resultData["expectedSurpluses"] = $expectedSurpluses;

$resultData["expectedDeficitLimit"] = $expectedDeficitLimit;

$resultData["expectedSurplusesLimit"] = $expectedSurplusesLimit;

$resultData["suitableStartIntervalIndex"] = $suitableStartIntervalIndex;

$resultData["suitableEndIntervalIndex"] = $suitableEndIntervalIndex;

return $resultData;

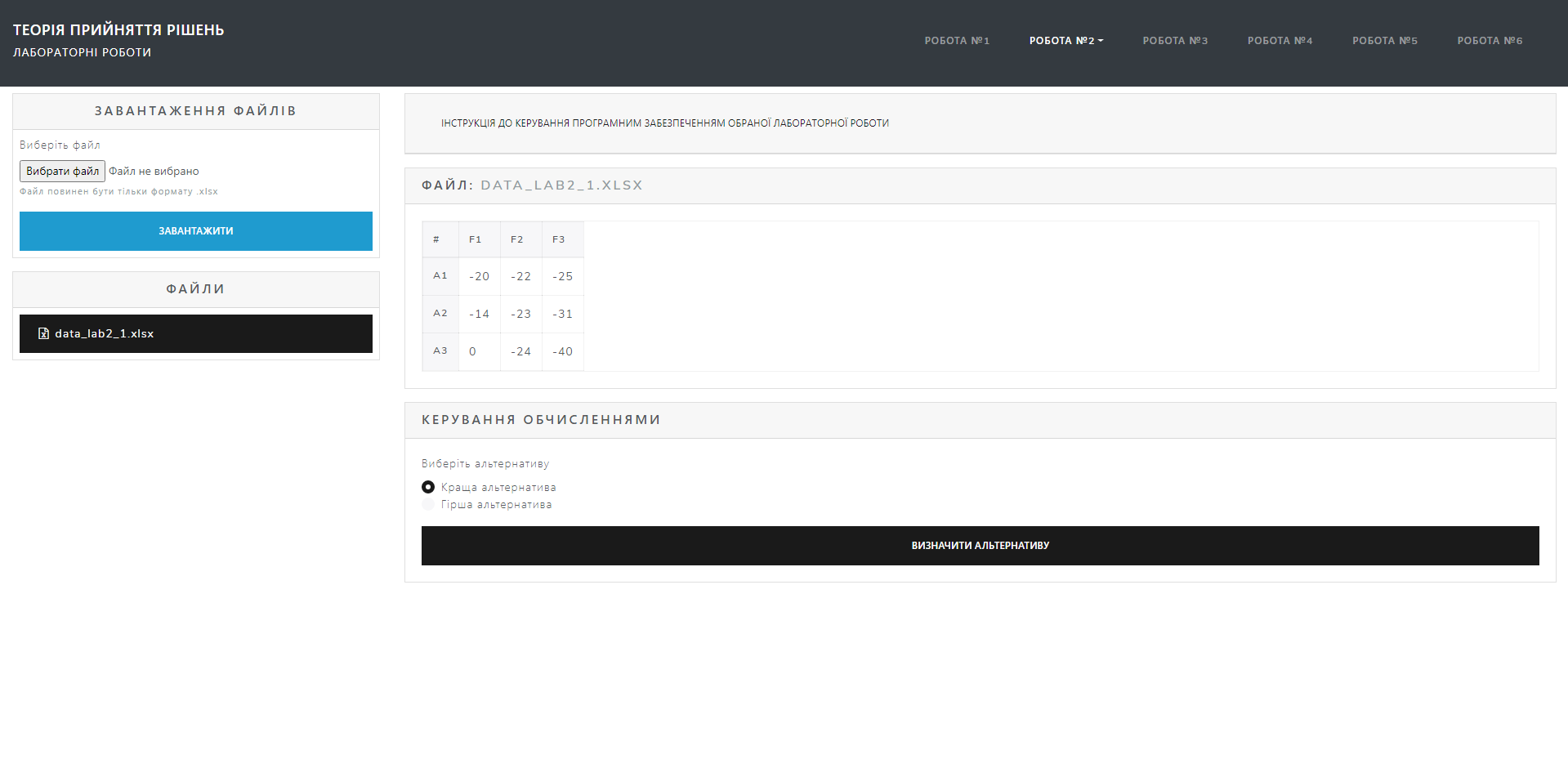
}

}

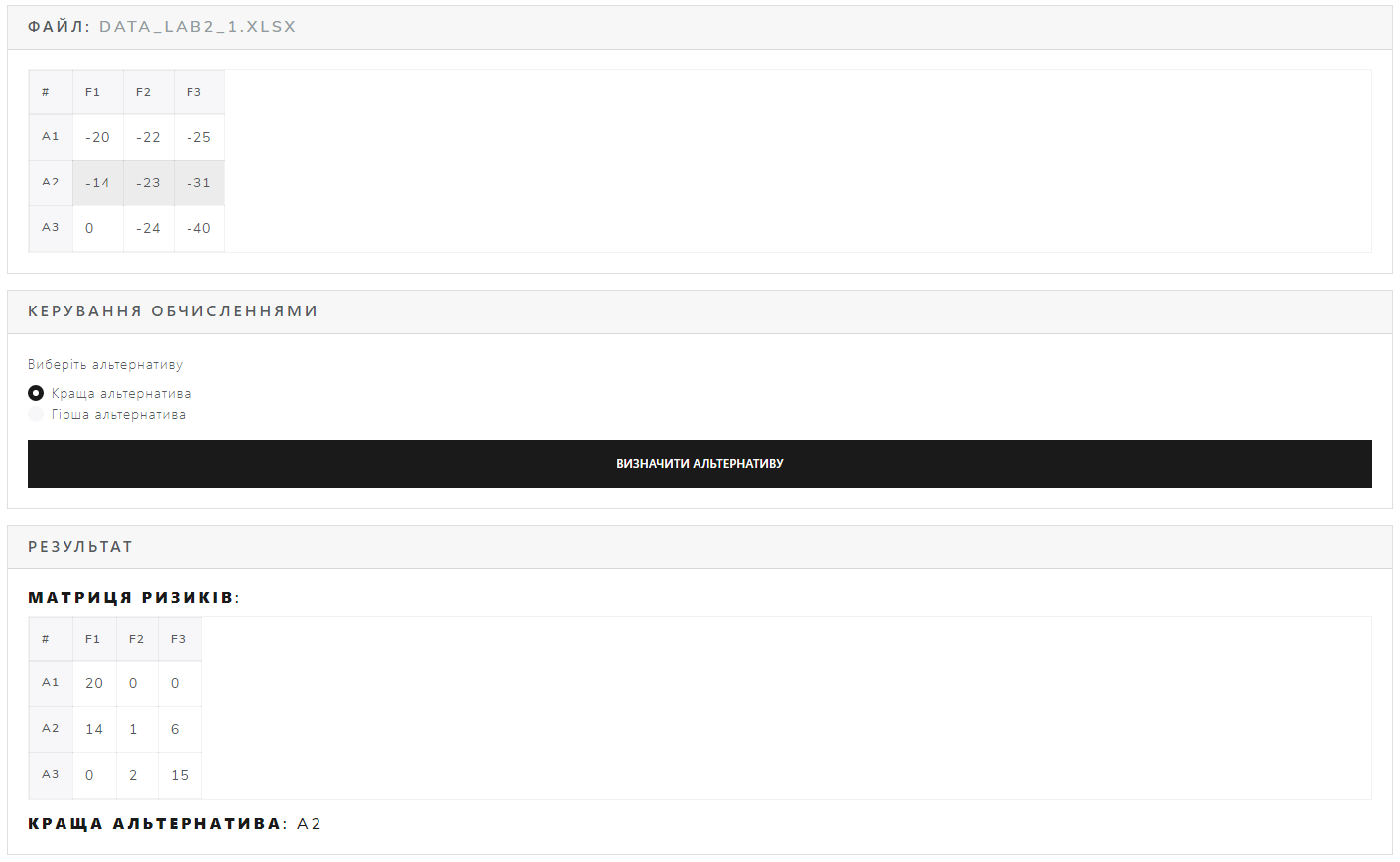
?>

**Результат роботи програми:**

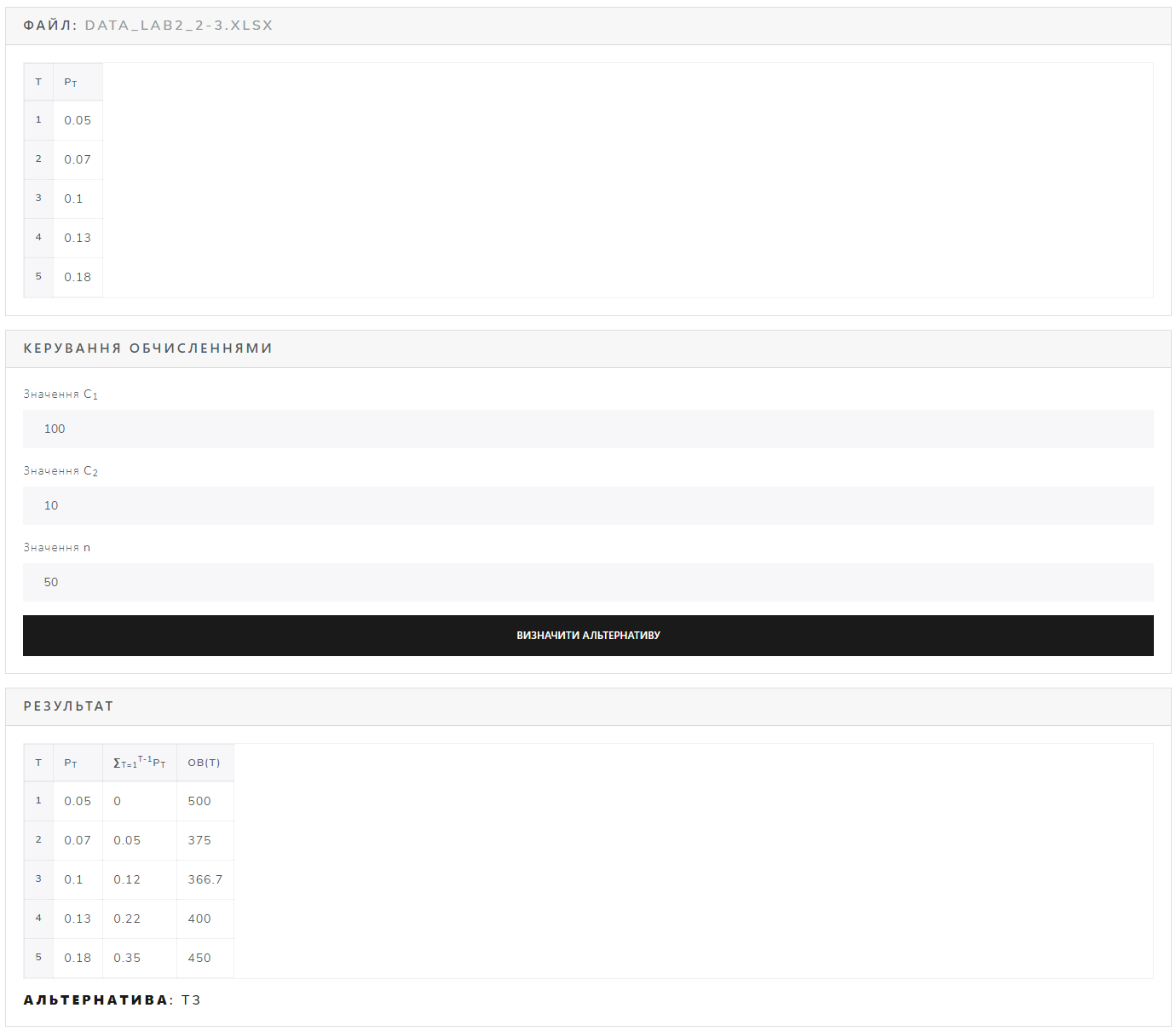
Для початку потрібно завантажити excel-файл із вхідними даними. Для цього потрібно скористатися формою «Завантаження файлів», що знаходиться зліва на робочому просторі програми. За допомогою вікна вибору обрати файл на комп’ютері та натиснути кнопку «Завантажити». Після завантаження, файл стане доступним у панелі «Файли», яка знаходиться в лівій частині робочого простору програми під формою завантаження файлів. Також, відразу після завантаження, в панелі «Файл», що знаходиться в правій частині робочого простору програми під інструкцією користувача, стануть доступними для перегляду завантажені вхідні дані у вигляді таблиці. Під нею є додаткова панель «Керування обчисленнями», що дозволяє виставляти різні параметри для проведення обрахунків. Додаткові інструкції з використання даного програмного забезпечення наведені в панелі «Інструкція до керування програмним забезпеченням обраної лабораторної роботи».



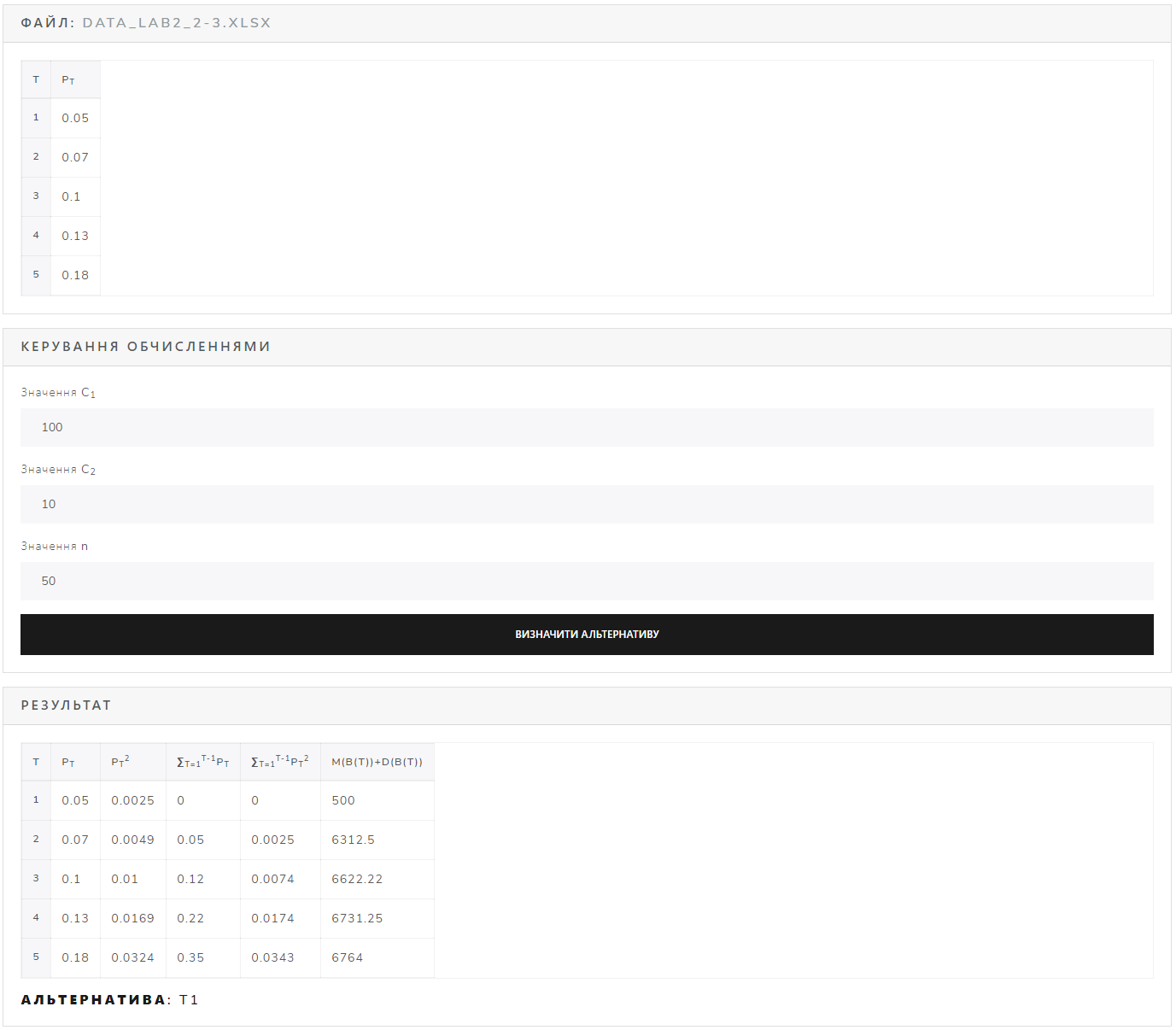
*Рис. 2.1. Вигляд робочого простору після завантаження вхідних даних*



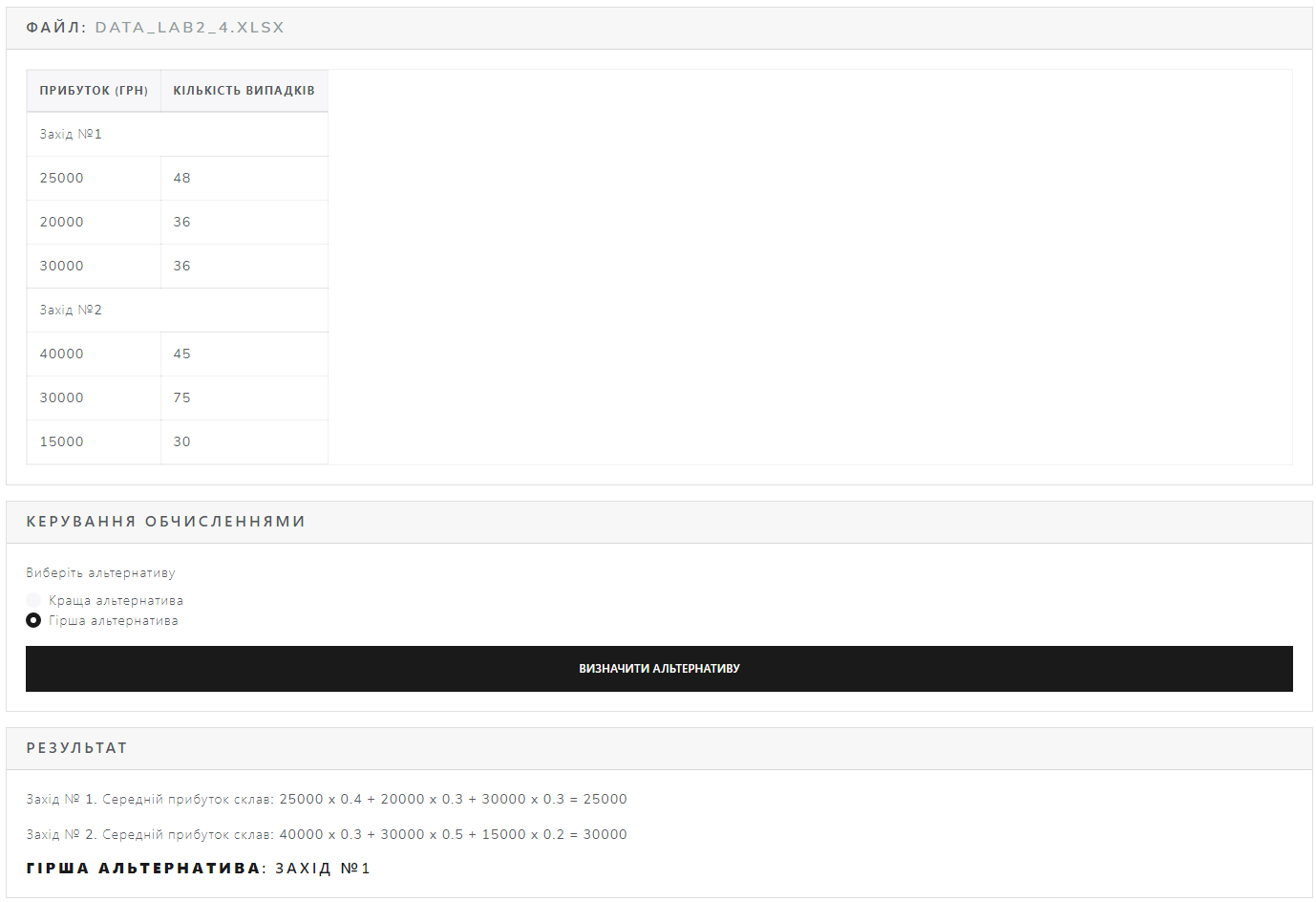
*Рис. 2.2. Визначення кращої альтернативи для критерію мінімаксного ризику Севіджа*



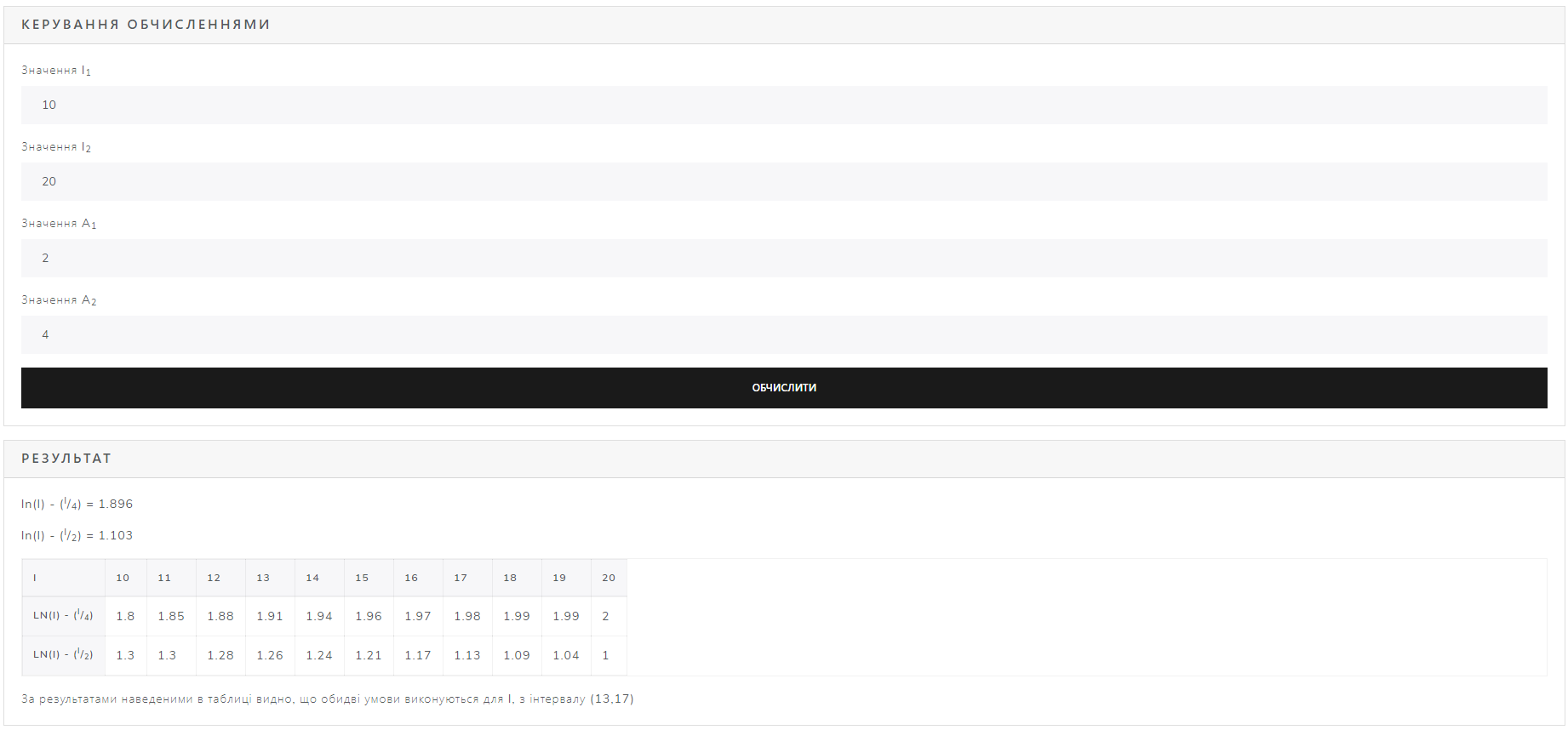
*Рис. 2.3. Визначення кращої альтернативи для критерію очікуваного значення*



*Рис. 2.4. Визначення кращої альтернативи для критерію «очікуване значення – дисперсія»*



*Рис. 2.5. Визначення гіршої альтернативи для критерію середнього очікуваного значення*



*Рис. 2.6. Визначення інтервалу для критерію граничного рівня*

**Висновок**: під час виконання цієї лабораторної роботи я провів вибір рішення в умовах ризику. Засвоїв навички аналізу альтернатив в умовах ризику. Ознайомився з критеріями ухвалення рішень в умовах ризику: критерієм Севіджа, критерієм очікуваного значення, критерієм середнього очікуваного значення, критерієм комбінації очікуваного значення і дисперсії та критерієм відомого граничного рівня. Створив web-програму для визначення альтернативи в умовах ризику на основі вхідних даних у форматі excel-файлу.

# **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:**

<?

namespace App\TPR;

class Lab3 {

public $tprSpreadsheet = array();

public function \_\_construct($data) {

$this->tprSpreadsheet = $data;

}

public function summationOfRanksCalculate($data, $bestAlternative = true) {

$resultData = array();

$ranksArray = array();

foreach ($data as $row) {

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

if ($index != 0) {

if (!isset($ranksArray[$column])) {

$ranksArray[$column] = array();

}

array\_push($ranksArray[$column], $index);

}

}

}

$value = array\_sum(array\_values($ranksArray)[0]);

$alternative = array\_keys($ranksArray)[0];

foreach ($ranksArray as $index => $rank) {

if ($bestAlternative) {

if (array\_sum($rank) < $value) {

$value = array\_sum($rank);

$alternative = $index;

}

} else {

if (array\_sum($rank) > $value) {

$value = array\_sum($rank);

$alternative = $index;

}

}

}

$resultData["value"] = $value;

$resultData["alternative"] = $alternative;

$resultData["ranksArray"] = $ranksArray;

return $resultData;

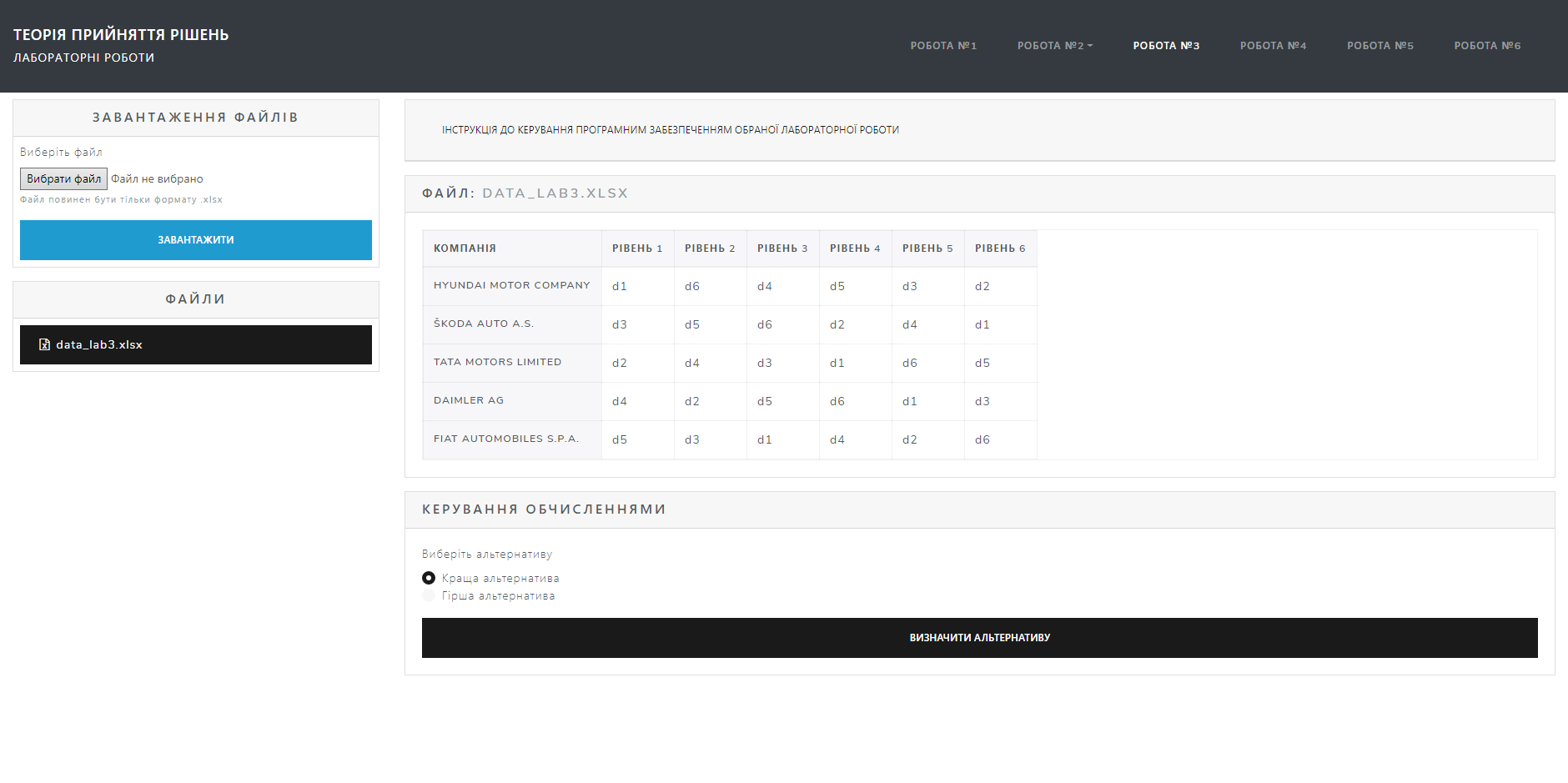
}

}

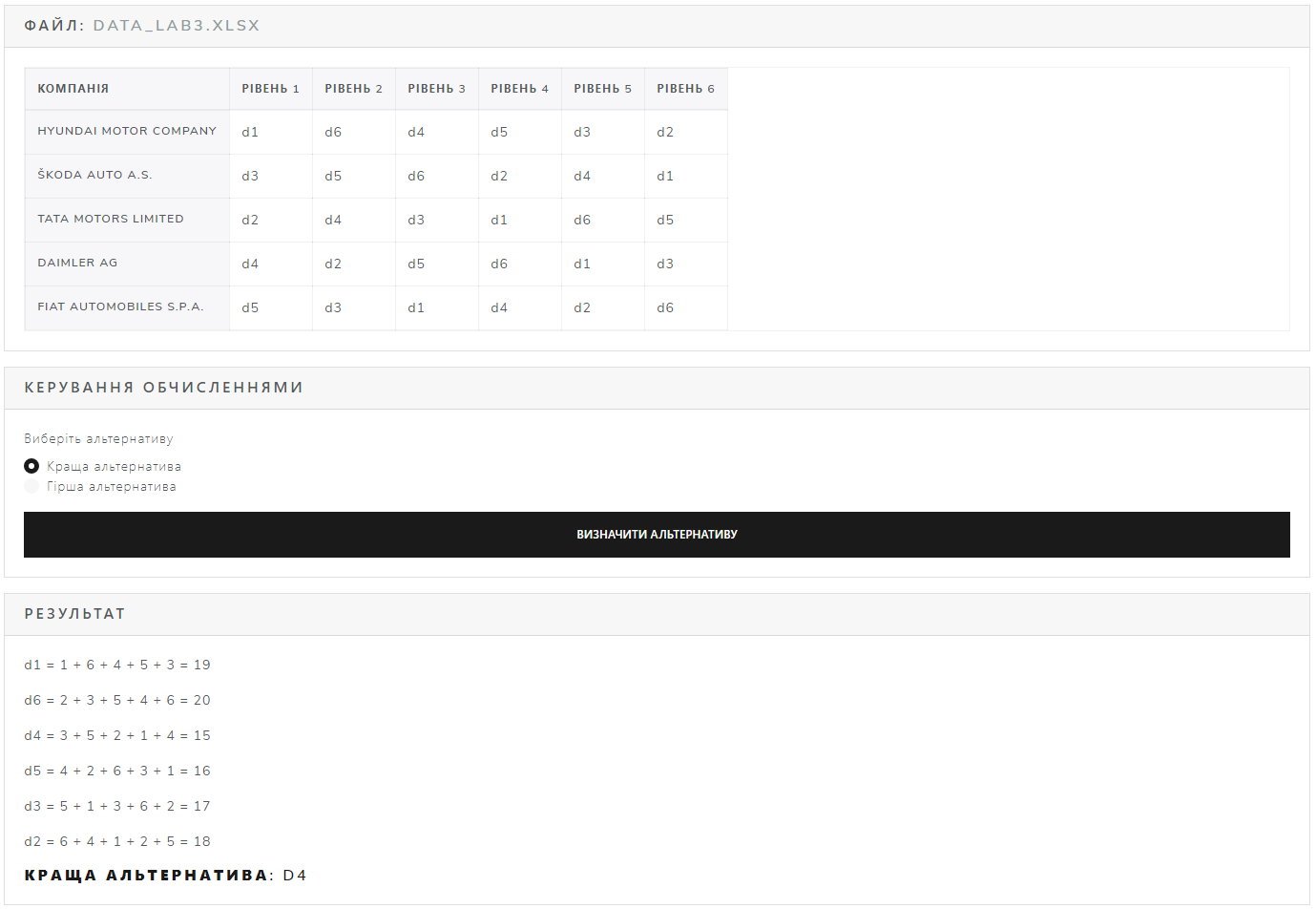
?>

**Результат роботи програми:**

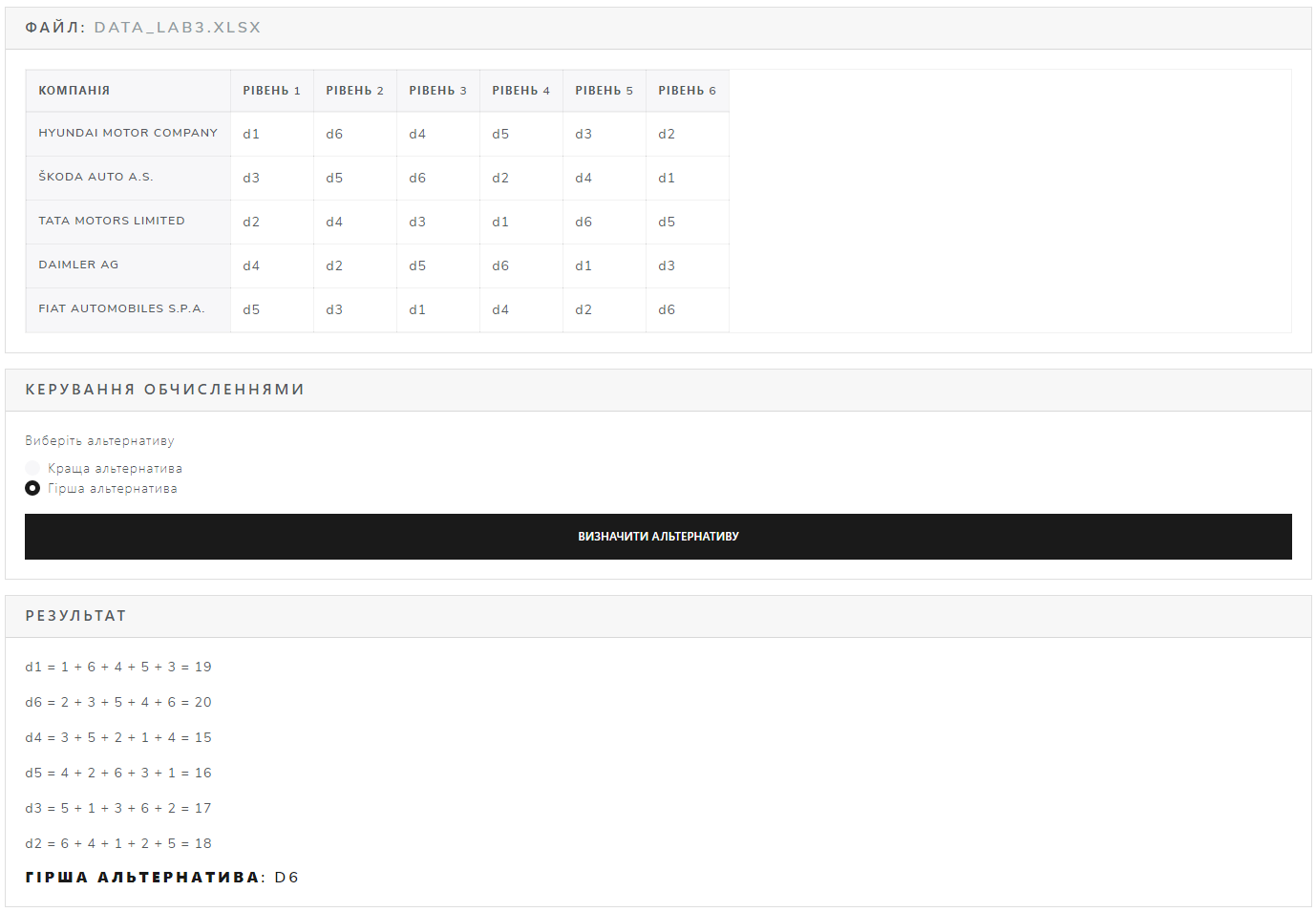
Для початку потрібно завантажити excel-файл із вхідними даними. Для цього потрібно скористатися формою «Завантаження файлів», що знаходиться зліва на робочому просторі програми. За допомогою вікна вибору обрати файл на комп’ютері та натиснути кнопку «Завантажити». Після завантаження, файл стане доступним у панелі «Файли», яка знаходиться в лівій частині робочого простору програми під формою завантаження файлів. Також, відразу після завантаження, в панелі «Файл», що знаходиться в правій частині робочого простору програми під інструкцією користувача, стануть доступними для перегляду завантажені вхідні дані у вигляді таблиці. Під нею є додаткова панель «Керування обчисленнями», що дозволяє виставляти різні параметри для проведення обрахунків. Додаткові інструкції з використання даного програмного забезпечення наведені в панелі «Інструкція до керування програмним забезпеченням обраної лабораторної роботи».



*Рис. 3.1. Вигляд робочого простору після завантаження вхідних даних*



*Рис. 3.2. Визначення кращої альтернативи для групового рішення*



*Рис. 3.3. Визначення гіршої альтернативи для групового рішення*

**Висновок**: під час виконання цієї лабораторної роботи я застосував метод аналізу при прийнятті групових рішень. Закріпив навички використання методу аналізу при прийнятті групових рішень. Створив web-програму для прийняття групового рішення на основі вхідних даних у форматі excel-файлу. Прийняв рішення, яке ґрунтується на порівнянні кількох можливих альтернативних рішень та виборі одного рішення, яке найбільш відповідає висунутим критеріям, використовуючи створену програму.

# **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:** Метод аналізу ієрархій (МАІ) – це математична процедура для ієрархічного зображення елементів з метою визначення суті будь-якої проблеми.

Метод полягає в декомпозиції проблеми на простіші складові частини, а також в обробленні суджень особи чи осіб, котрі приймають рішення (ОПР) на підставі парних порівнянь пріоритетів (критеріїв) доцільності. Це дає змогу оцінити рівень взаємодії елементів ієрархії.

МАІ включає декілька етапів:

* перший етап – структуризація проблеми у вигляді ієрархії;
* другий етап – визначення локальних пріоритетів (критеріїв) та оцінювання кожної з альтернатив. Елементи задачі порівнюються попарно стосовно їх впливу на загальну для них характеристику;
* третій етап – побудова матриці попарних порівнянь на основі принципу дискримінації і порівнюваності суджень;
* четвертий етап – визначення параметрів матриці пріоритетів;
* п’ятий етап – узгодженість локальних пріоритетів;
* шостий етап – складання матриць попарних порівнянь за варіантами за кожним елементом квадратної матриці;
* сьомий етап – визначення глобальних (загальних) пріоритетів.

Загальна структура методу аналізу ієрархій може включати кілька ієрархічних рівнів із своїми критеріями.

Складність методу аналізу ієрархій полягає у визначенні відносних вагових коефіцієнтів для оцінки альтернативних рішень. Якщо мається n критеріїв на заданому рівні ієрархії, що відповідає процедура створює матрицю A розмірності n×n, іменовану матрицею парних порівнянь, що відбиває судження обличчя, що приймає рішення, щодо важливості різних критеріїв. Парне порівняння виконаються таким чином, що критерій у рядку i (i = 1, 2, ..., n) оцінюється щодо кожного з критеріїв, представлених n стовпцями. Позначимо через aij елемент матриці A, 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 aij = 1 означає, что i-й і j-й критерії однаково важливі, aij = 5 відбиває думку, що i-й ритерій значно важливіше, ніж k-й, а aij = 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 aij = k, то автоматично aji= 1/k. Крім того, усі діагональні елементи аii матриці А повинні бути рівні 1, тому що вони виражають оцінку критерію щодо самих себе.

**Код програми:**

<?

namespace App\TPR;

class Lab4 {

public $tprSpreadsheet = array();

public function \_\_construct($data) {

$this->tprSpreadsheet = $data;

}

public function analysisOfHierarchiesCalculate($data, $bestAlternative = true) {

$resultData = array();

$strategyArray = array();

$strategySum = array();

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

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

if ($index > 1) {

if (!isset($strategyArray[$index - 2])) {

$strategyArray[$index - 2] = array();

}

$additionElement = array();

array\_push($additionElement, $column / 100);

array\_push($additionElement, $data[$key][1] / 100);

array\_push($strategyArray[$index - 2], $additionElement);

}

}

}

$value = 0;

$alternative = 0;

foreach ($strategyArray as $index => $strategy) {

$additionSum = 0;

foreach ($strategy as $additionElement) {

$additionSum += $additionElement[0] \* $additionElement[1];

}

array\_push($strategySum, $additionSum);

if ($index == 0) {

$value = $additionSum;

}

if ($bestAlternative) {

if ($additionSum > $value) {

$value = $additionSum;

$alternative = $index;

}

} else {

if ($additionSum < $value) {

$value = $additionSum;

$alternative = $index;

}

}

}

$resultData["value"] = $value;

$resultData["alternative"] = $alternative;

$resultData["strategyArray"] = $strategyArray;

$resultData["strategySum"] = $strategySum;

return $resultData;

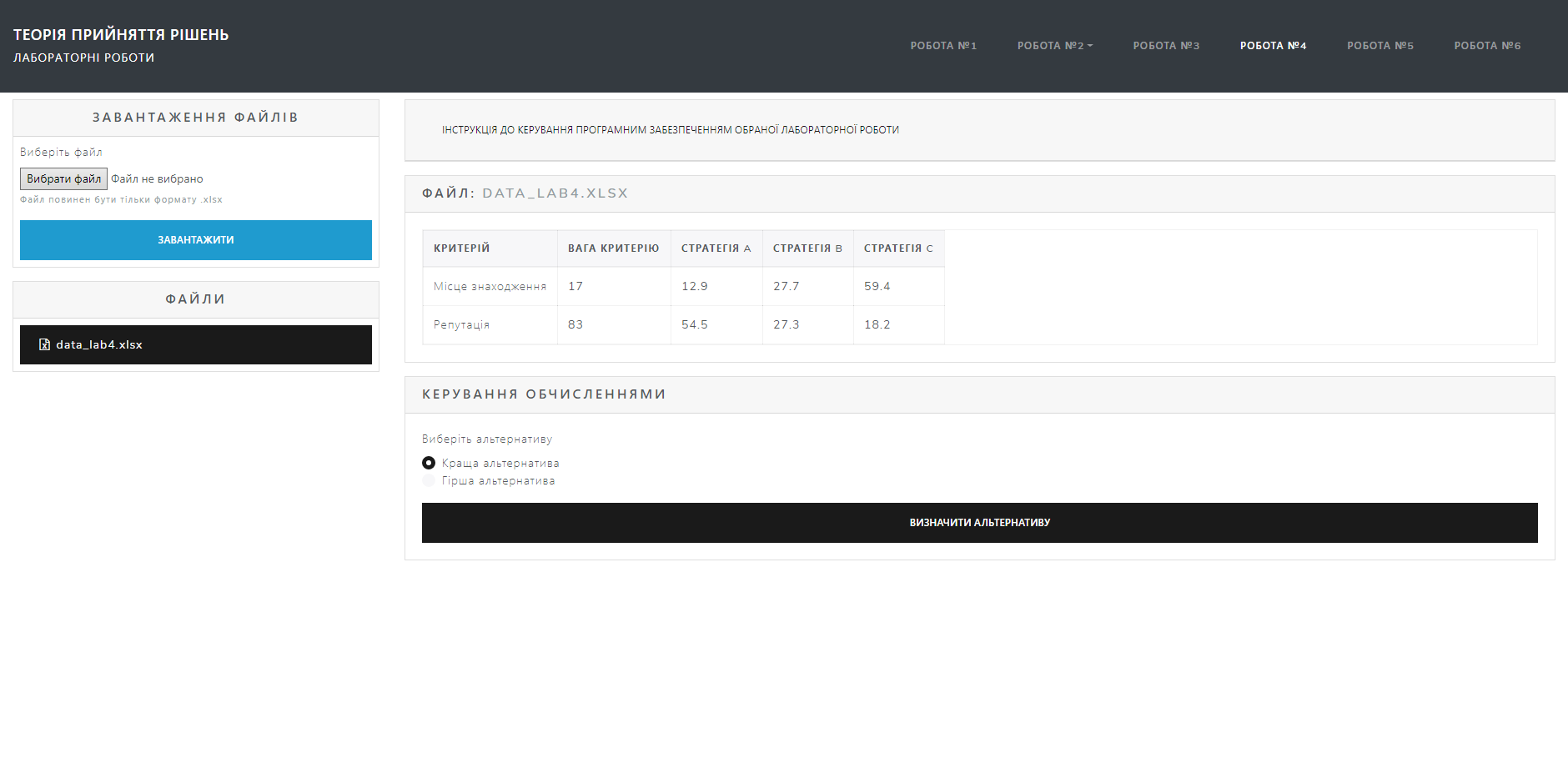
}

}

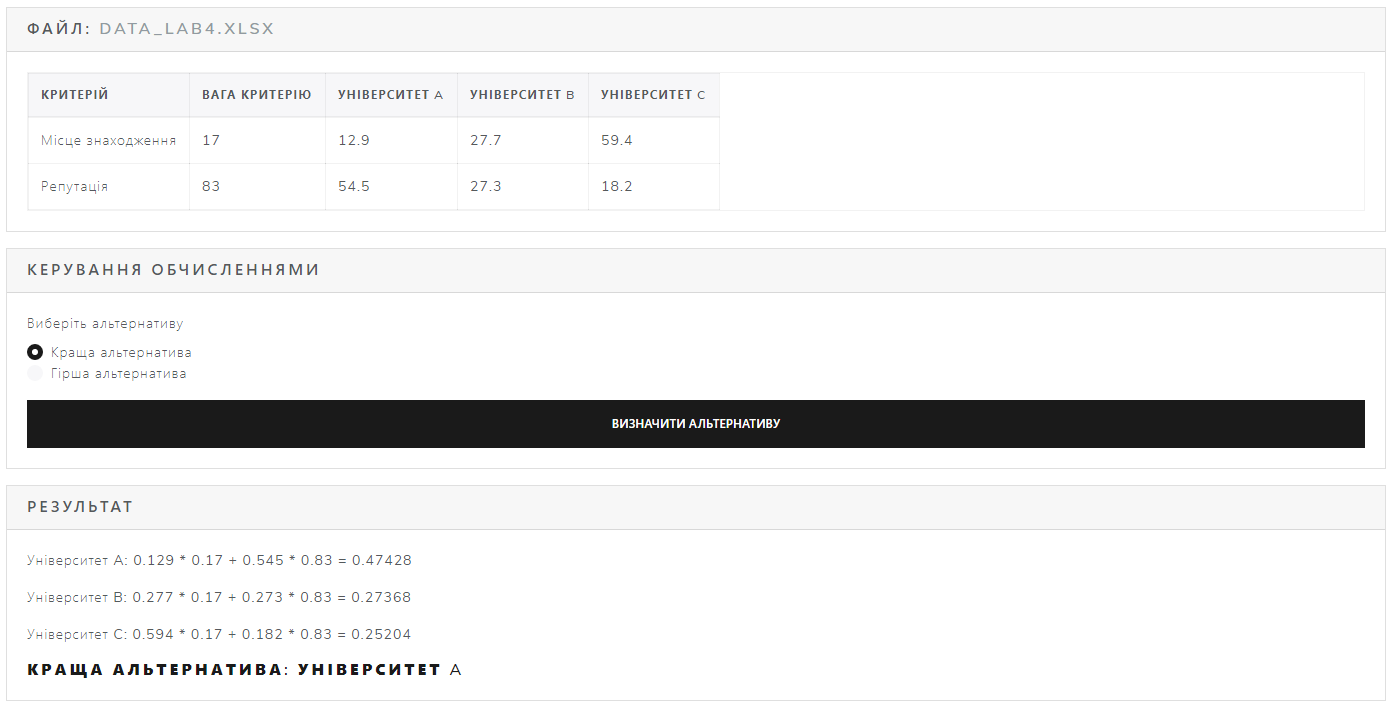
?>

**Результат роботи програми:**

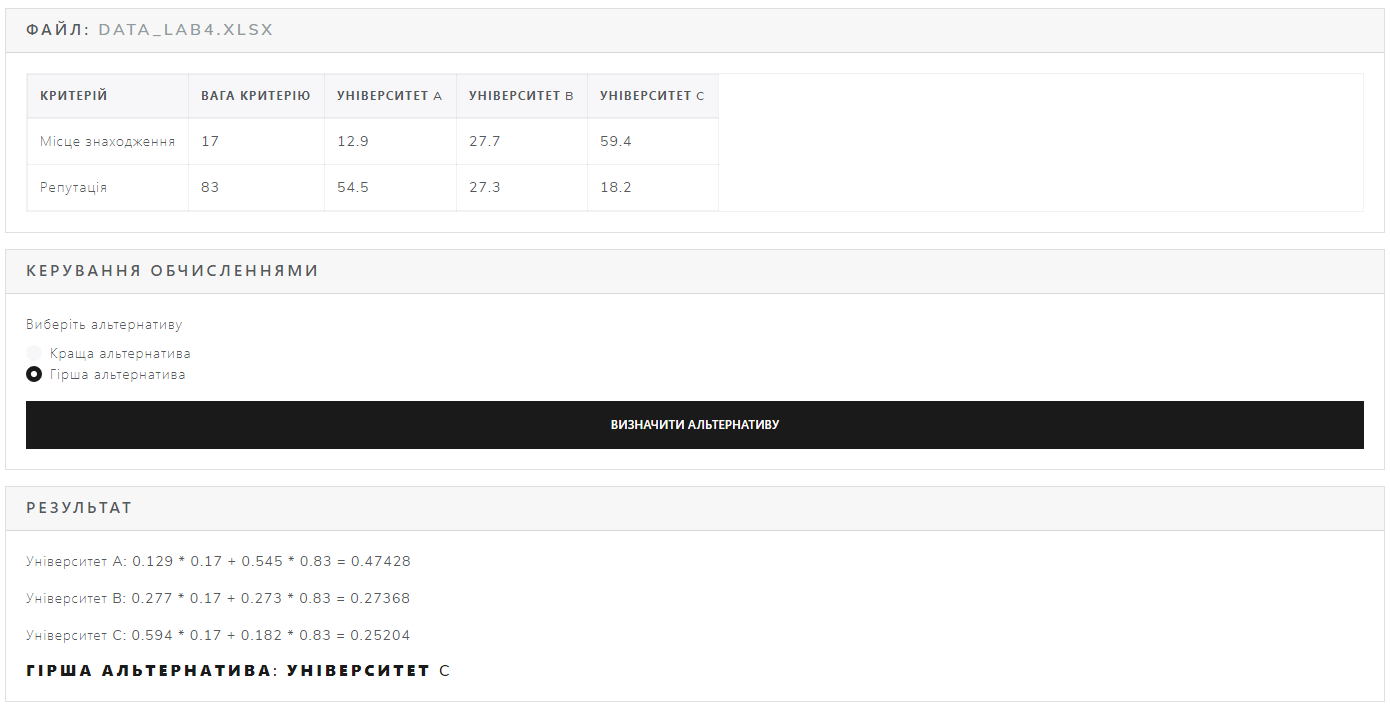
Для початку потрібно завантажити excel-файл із вхідними даними. Для цього потрібно скористатися формою «Завантаження файлів», що знаходиться зліва на робочому просторі програми. За допомогою вікна вибору обрати файл на комп’ютері та натиснути кнопку «Завантажити». Після завантаження, файл стане доступним у панелі «Файли», яка знаходиться в лівій частині робочого простору програми під формою завантаження файлів. Також, відразу після завантаження, в панелі «Файл», що знаходиться в правій частині робочого простору програми під інструкцією користувача, стануть доступними для перегляду завантажені вхідні дані у вигляді таблиці. Під нею є додаткова панель «Керування обчисленнями», що дозволяє виставляти різні параметри для проведення обрахунків. Додаткові інструкції з використання даного програмного забезпечення наведені в панелі «Інструкція до керування програмним забезпеченням обраної лабораторної роботи».



*Рис. 4.1. Вигляд робочого простору після завантаження вхідних даних*



*Рис. 4.2. Визначення кращої альтернативи для методу аналізу ієрархій*



*Рис. 4.3. Визначення гіршої альтернативи для методу аналізу ієрархій*

**Висновок**: під час виконання цієї лабораторної роботи я використав метод аналізу ієрархій для вирішення задач системного аналізу. Сформував навички розпізнавання типів задач системного аналізу, які можуть бути вирішені з використанням методу аналізу ієрархій, які мають один або кілька критеріальних рівні. Створив web-програму для визначення альтернативи методом аналізу ієрархійдля вирішення задач системного аналізу на основі вхідних даних у форматі excel-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 різною метою.

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

Кожний гравець має певний набір 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, повязані 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» хі, а в стовбчиках - показники ЗВ (кj). В комірці ігрової матриці запишемо ефективності Εi,j.

Обидві матриці мають однакову кількість рядків, що відображають можливі стратегії «покриття» схеми ЗВ конструктивними модулями. Кількість стовпчиків 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: Еi,j(p)=Pi,j/ci, де Pi,j - ймовірність безвідмовної роботи сукупності модулів, що необхідні для реалізації схеми по і-й стратегії; сі - вартість модулів, що використовуються при і-й стратегії.

Показник ефективності ЗВ по швидкодії орієнтовно модна знайти, використавши співвідношення: Еi,j(t)=1/ci,ti, де ti - 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: Еi,j(w)=1/Wic, де Wi - енергія, що споживається всіма модулями при виборі елементної бази по і-й стратгії; с - коефіцієнт вартості одиниці енергії.

Показник ефективості ЗВ по кількості зовнішніх звзків модулів, що застосовуються по і-й стратегії, можуть характеризувати трудомісткість і надійність ЗВ. В задачах мінімізації цей показник визначається виразом: Еi,j(n)=1/Ni, де Ni - загальна кільеість зовнішніх виводів конструктивних модулів, що застосовуються у виробі по і-й стратегії.

Таблиця 1 – Початкова ігрова матриця

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Стратегія | K1 | . . . | Kj | . . . | Kn |
| x1 | E11 | . . . | E1j | . . . | E1n |
| . . . | . . . | . . . | . . . | . . . | . . . |
| xi | Ei1 | . . . | Eij | . . . | Ein |
| xm | Em1 | . . . | Eij | . . . | Emn |
| Emax | . . . | Ejmax | . . . | Enmax |

Виконаємо нормування простору пошуку. Всі елементи ігрової матриці виразимо в одних одиницях. Для цього виберем в кожному її стовбчику максимальне значення якості і розділимо на нього фактично отримане значення якості. В результаті прийдемо до матриці з безрозмірними елементами, які рівняються або менші за одиницю: aij=Eij/Ejmax або aij=Ejmin/Eij.

Складемо ігрову матрицю у вигляді таблиці 2.

Таблиця 2 – Ігрова матриця з нормованими елементами

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Стратегія | K1 | . . . | Kj | . . . | Kn |
| x1 | a11 | . . . | a1j | . . . | a1n |
| . . . | . . . | . . . | . . . | . . . | . . . |
| xi | ai1 | . . . | aij | . . . | ain |
| xm | am1 | . . . | aij | . . . | amn |

Показники якості ЗВ не є рівноправними, тому їх необхідно зпівставити (за допомогою експертного методу). При цьому кожний елемент матриці, що максимізується, необхідно помножити на відповідний ваговий коефіцієнт. В результаті отримаємо зважену ігрову матрицю [aij\*]nm у вигляді таблиці 3.

Таблиця 3 – Зважена ігрова матриця.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Стратегія | K1 | . . . | Kj | . . . | Kn | Сумарний виграш |
| x1 | a11\* | . . . | a1j\* | . . . | a1n\* | ∑ a1j\*=A1 |
| . . . | . . . | . . . | . . . | . . . | . . . | . . . |
| xm | a m1\* | . . . | a ij\* | . . . | a mn\* | ∑ amj\*=Am |

Аналогічно, помноживши на відповідні вагові коефіцієнти елементи другої ігрової матриці показників якості, що мінімізуються, знайдемо взвішену ігрову матрицю мінімізуємих параметрів [bij]fn. В ці ж таблиці занесли згортку у вигляді суми параметрів, що характеризують сумарний виграш по стратегіям.

Для визначення оптимальної стратегії необхідно:

1. 1)підрахувати сумарні виграші по кожній із стратегій в таблиці 3;
2. 2)оцінити кожну із відповідних стратегій параметрів ЗВ, що максимізуютьсяі мінімізуються, за допомогою виразу. В чисельнику знаходяться параметри, що максимізуються, а в знаменнику – ті, що мінімізуються:

Оптимальною буде та стратегія, яка дає максимальне значення виграша E(xi).

**Код програми:**

<?

namespace App\TPR;

class Lab5 {

public $tprSpreadsheet = array();

public function \_\_construct($data) {

$this->tprSpreadsheet = $data;

}

public function theoryOfGameCalculate($data, $coefficient, $maximizeParams, $bestAlternative = true) {

$resultData = array();

$efficiencyMatrix = array();

$normalizeMatrix = array();

$weightCoefficientMatrix = array();

$valuesByParams = array();

$minOrMaxvaluesByParams = array();

$sumOfMinimizeParams = array();

$sumOfMaximizeParams = array();

$efficiencyMatrix = $data;

foreach ($efficiencyMatrix as $key => $row) {

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

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

if ($index != 0) {

switch ($index) {

case 1:

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

break;

case 2:

$efficiencyMatrix[$key][$index] = $efficiencyMatrix[$key][$index] / $Ec;

break;

case 3:

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

break;

case 4:

$efficiencyMatrix[$key][$index] = $efficiencyMatrix[$key][$index] / $Ec;

break;

case 5:

break;

}

}

}

}

foreach ($efficiencyMatrix as $key => $row) {

if ($key == 0) {

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

array\_push($valuesByParams, array());

}

}

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

if ($index != 0) {

array\_push($valuesByParams[$index - 1], $column);

}

}

}

foreach ($valuesByParams as $key => $row) {

$minOrMaxValues = $row[0];

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

if ($maximizeParams[$key]) {

if ($column > $minOrMaxValues)

$minOrMaxValues = $column;

} else {

if ($column < $minOrMaxValues)

$minOrMaxValues = $column;

}

}

array\_push($minOrMaxvaluesByParams, $minOrMaxValues);

}

$normalizeMatrix = $efficiencyMatrix;

foreach ($normalizeMatrix as $key => $row) {

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

if ($index != 0) {

if ($maximizeParams[$key]) {

$normalizeMatrix[$key][$index] /= $minOrMaxvaluesByParams[$index - 1];

} else {

$normalizeMatrix[$key][$index] = $minOrMaxvaluesByParams[$index - 1] / $efficiencyMatrix[$key][$index];

}

}

}

}

$weightCoefficientMatrix = $normalizeMatrix;

foreach ($weightCoefficientMatrix as $key => $row) {

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

if ($index != 0) {

$weightCoefficientMatrix[$key][$index] = $weightCoefficientMatrix[$key][$index] \* $coefficient[$index - 1];

}

}

}

foreach ($weightCoefficientMatrix as $key => $row) {

$sumOfMax = 0;

$sumOfMin = 0;

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

if ($index != 0) {

if ($maximizeParams[$index - 1]) {

$sumOfMax += $column;

} else {

$sumOfMin += $column;

}

}

}

array\_push($sumOfMaximizeParams, $sumOfMax);

array\_push($sumOfMinimizeParams, $sumOfMin);

}

$value = $sumOfMaximizeParams[0] / $sumOfMinimizeParams[0];

$alternative = 0;

for ($i = 0; $i < count($weightCoefficientMatrix); $i++) {

if ($bestAlternative) {

if ($sumOfMaximizeParams[$i] / $sumOfMinimizeParams[$i] > $value) {

$value = $sumOfMaximizeParams[$i] / $sumOfMinimizeParams[$i];

$alternative = $i;

}

} else {

if ($sumOfMaximizeParams[$i] / $sumOfMinimizeParams[$i] < $value) {

$value = $sumOfMaximizeParams[$i] / $sumOfMinimizeParams[$i];

$alternative = $i;

}

}

}

$resultData["value"] = $value;

$resultData["alternative"] = $alternative;

$resultData["efficiencyMatrix"] = $efficiencyMatrix;

$resultData["normalizeMatrix"] = $normalizeMatrix;

$resultData["weightCoefficientMatrix"] = $weightCoefficientMatrix;

$resultData["minOrMaxvaluesByParams"] = $minOrMaxvaluesByParams;

$resultData["sumOfMinimizeParams"] = $sumOfMinimizeParams;

$resultData["sumOfMaximizeParams"] = $sumOfMaximizeParams;

return $resultData;

}

}

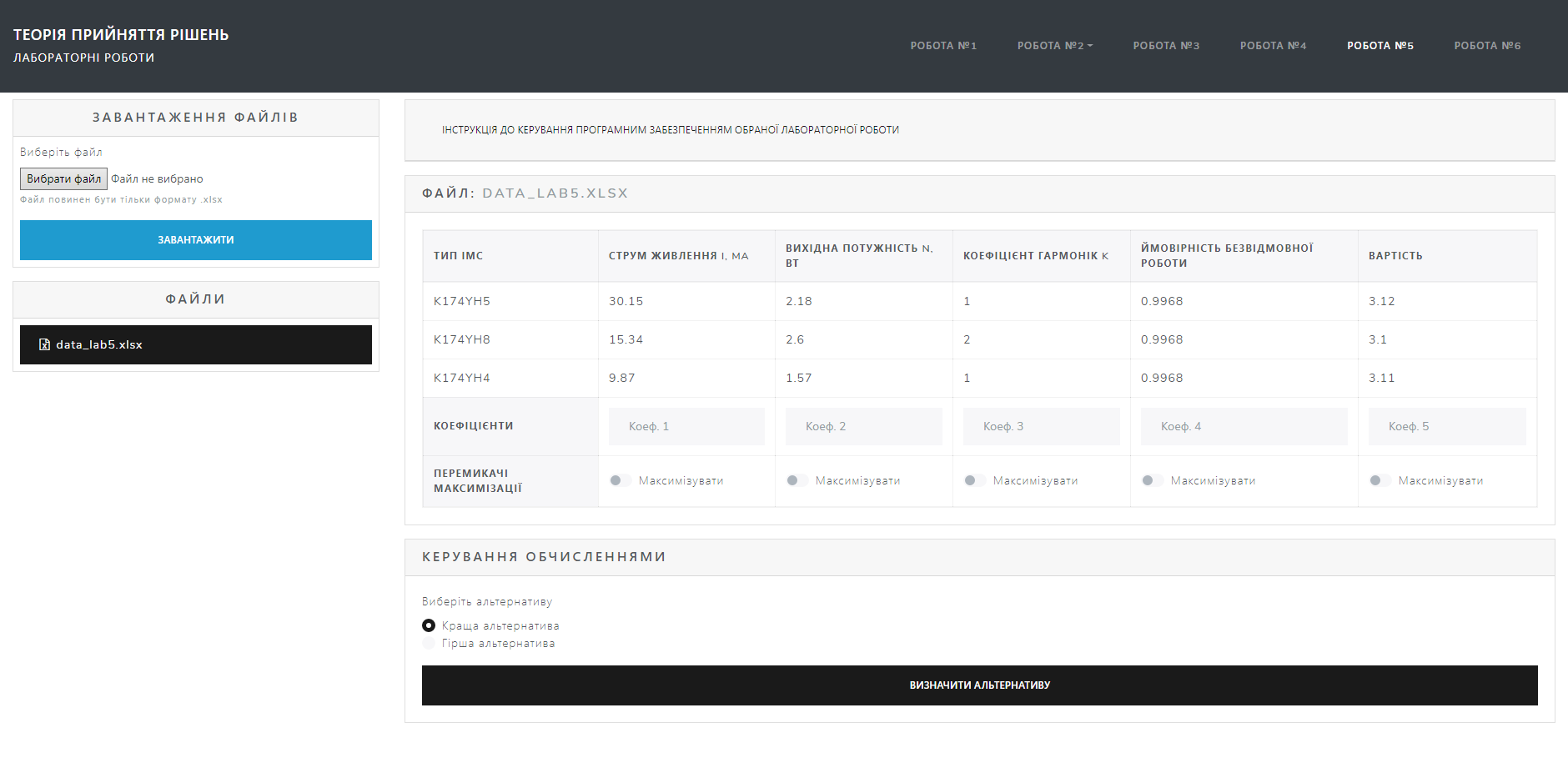
?>

**Результат роботи програми:**

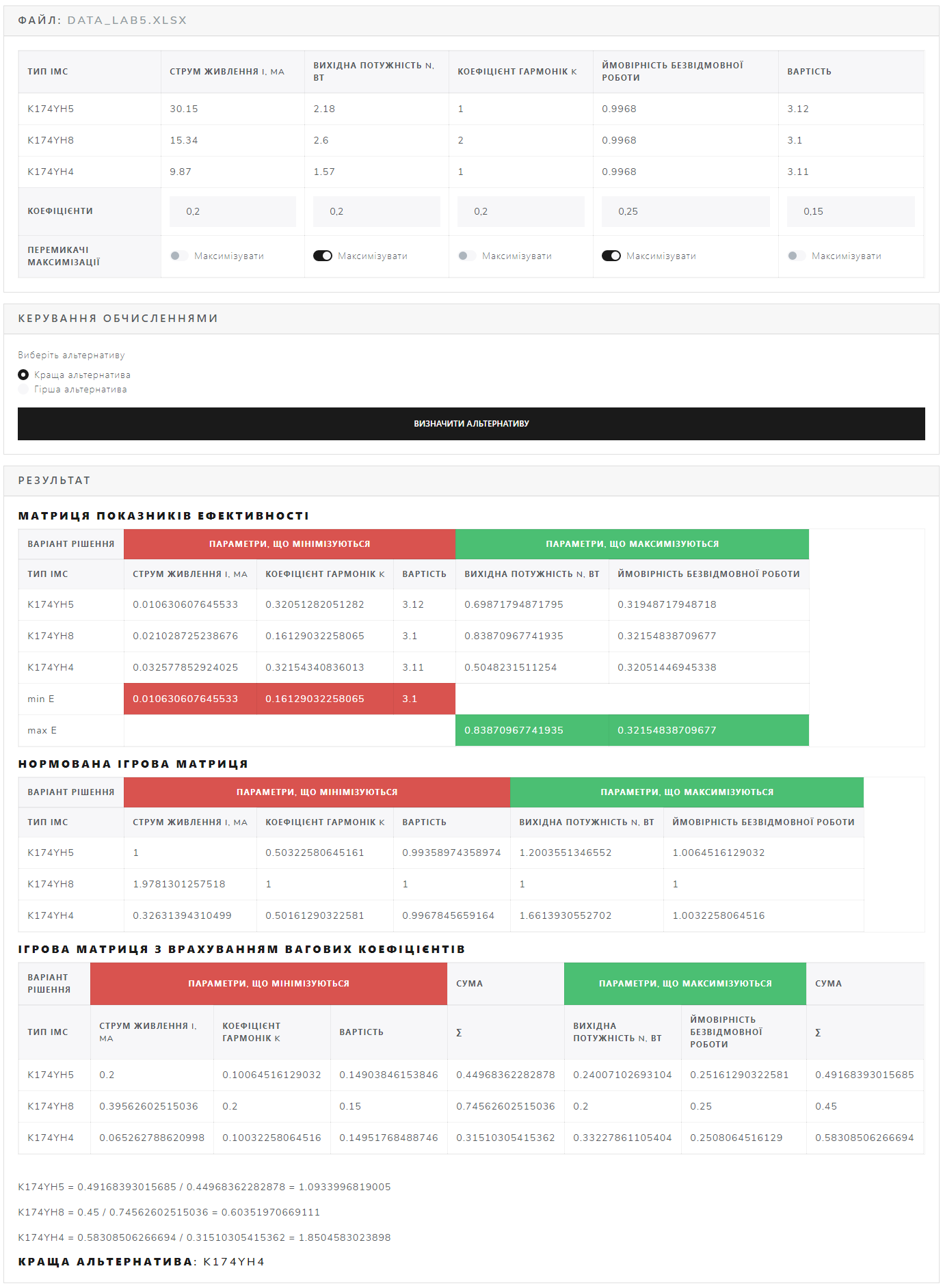
Для початку потрібно завантажити excel-файл із вхідними даними. Для цього потрібно скористатися формою «Завантаження файлів», що знаходиться зліва на робочому просторі програми. За допомогою вікна вибору обрати файл на комп’ютері та натиснути кнопку «Завантажити». Після завантаження, файл стане доступним у панелі «Файли», яка знаходиться в лівій частині робочого простору програми під формою завантаження файлів. Також, відразу після завантаження, в панелі «Файл», що знаходиться в правій частині робочого простору програми під інструкцією користувача, стануть доступними для перегляду завантажені вхідні дані у вигляді таблиці. В таблиці із вхідними даними з файлу в поля ряду "Коефіцієнти" потрібно ввести під кожним параметром коефіцієнт цього параметра. Сума всіх введених коефіцієнтів повинна дорівнювати 1. В таблиці із вхідними даними з файлу в ряді "Перемикачі максимізації" потрібно відмітити, які параметри потрібно максимізувати. Необхідно, щоб хоча б один параметр максимізувався і щоб хоча б один параметр мінімізувався. Під нею є додаткова панель «Керування обчисленнями», що дозволяє виставляти різні параметри для проведення обрахунків. Додаткові інструкції з використання даного програмного забезпечення наведені в панелі «Інструкція до керування програмним забезпеченням обраної лабораторної роботи».

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Варіант | К174УН5 | | К174УН8 | | К174УН4 | |
| Струм живлення I, мА | Вихідна потужність (при R=4 Ом) N, Вт | Струм живлення I, мА | Вихідна потужність  (при R=4 Ом) N, Вт | Струм живлення I, мА | Вихідна потужність (при R=4 Ом) N, Вт |
| 5 | 30.15 | 2.18 | 15.34 | 2.60 | 9.87 | 1.57 |

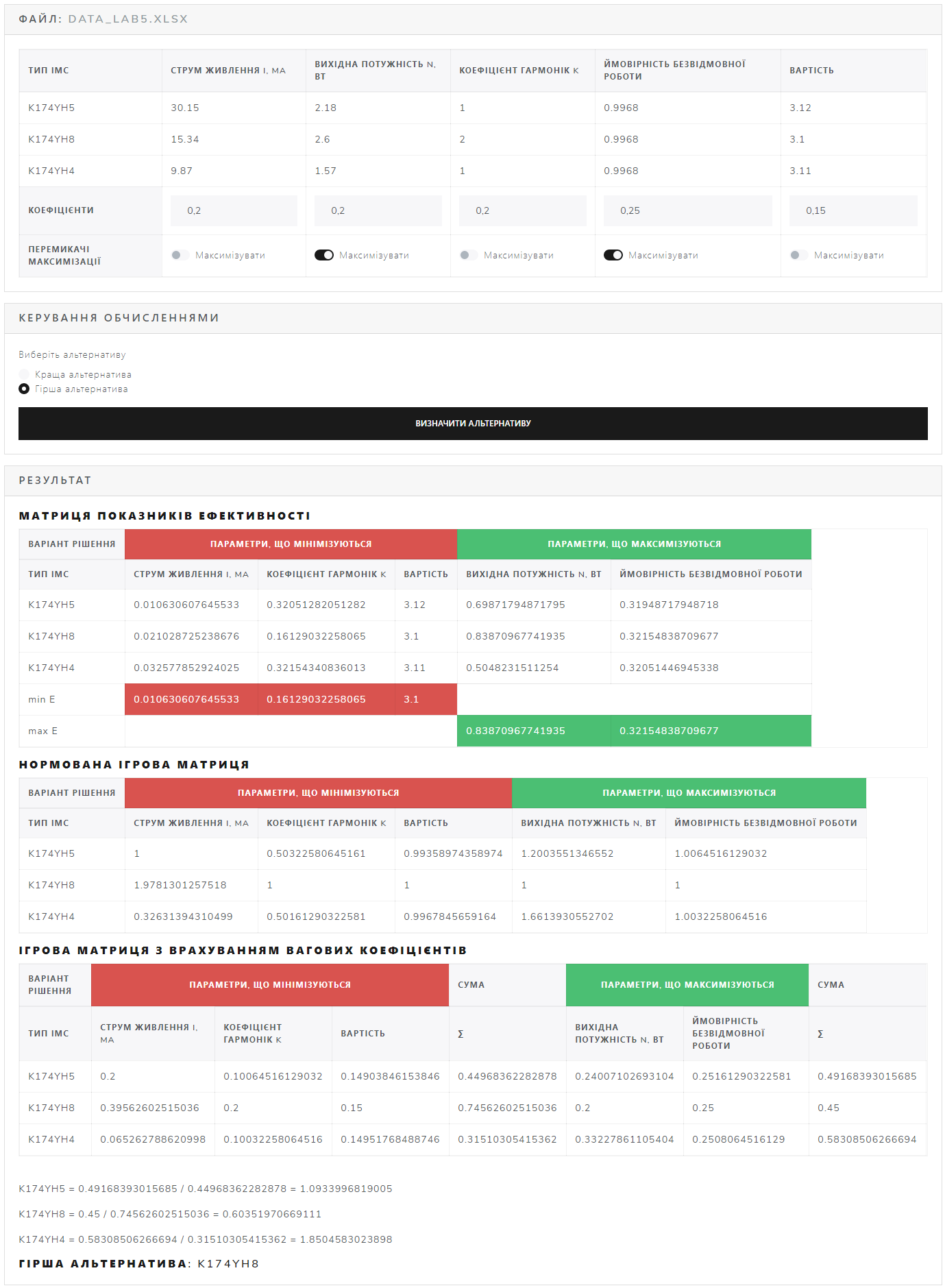
*Таблиця 5.1. Таблиця із завданням для обчислень відповідно до варіанту*



*Рис. 5.1. Вигляд робочого простору після завантаження вхідних даних*



*Рис. 5.2. Матриці та вагові коефіцієнти при пошуку крашої альтернативи*



*Рис. 5.3. Матриці та вагові коефіцієнти при пошуку гіршої альтернативи*

**Висновок**: під час виконання цієї лабораторної роботи я навчився використовувати теорії ігор при прийнятті рішень на основі груп конфліктних критеріїв. Закріпив навички використання теорії ігор при оптимізації елементної бази засобів вимірювання (ЗВ). Створив web-програму для визначення альтернативи, за допомогою теорії ігор на основі груп конфліктних критеріїв, на основі вхідних даних у форматі excel-файлу. Виконав пошук кращої та гіршої альтератив в створеній програмі, використовуючи вхідні дані відповідно до варіанту.

# **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 турніру. Той порядок визначення переможця (лідера) і розподіл місць сред інших учасників туріру, 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 (x1, x2, ..., xn) відповідає вершина графа. Якщо гравець хі виграв у гравця хj, 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.



Рисунок 1. – Граф результата змагань

Метод вирішення задачі реалізується таким чином. Будується матриця :



При цьому

,

де  Xi > Xj – означає виграш і-го гравця у j-го; Xi < Xj – означає програш і-го j-му; Xi = Xj – нічийний результат.

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



Розподіл балів серед гравців задаєься вектором: Р(1)=[P1(1), P2(1), ..., Pn(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)= [P1(2), P2(2), ..., Pn(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 Рі0(k) нормовану ітеровану силу k-го порядка і-го гравця:

В загальному вигляді процес розрахунку нормованої ітерованої «сили» гравців можна охарактеризувати таким співвідношенням:

 де  - сума компонент вектора АР(k-1).

Якщо матриця А не може бути розложена, то розглянута процедура, відповідно до теореми Перрона-Фробениуса, приводить до максимального власного числа матриці А з відповідним власним векторо 

Таким чином процес обчислювання нормованої іерованої «сили» гравців сходиться. На відміну від звичайного підрахунку балів описана процедура дозволяє врахувати опосередковані переваги гравців.

Розглянемо приклад розрахунку нормованої ітерованої «сили» пяти гравців. Результат турніра поданий системою порівнянь:

Х1>X2; X2>X4;

Х1>X3; X2>X5;

Х1=X4; X3<X4;

Х1>X5; X3<X5;

Х2>X3; X4>X5.

Приймемо, що  та складаємо квадратну матрицю суміжності, подану в таблиці 1. Розрахунок 1-ї ітерації поданий також в таблиці 1.

Таблиця 1. Квадратна матриця суміжності.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i \ j | X1 | X2 | X3 | X4 | X5 |  | Pi0(1) | Pi(2) | Pi0(2) |
| X1 | 1 | 2 | 2 | 1 | 2 | 8 | 0.32 | 36 | 0.395 |
| X2 | 0 | 1 | 2 | 2 | 2 | 7 | 0.28 | 27 | 0.297 |
| X3 | 0 | 0 | 1 | 0 | 0 | 1 | 0.04 | 1 | 0.011 |
| X4 | 1 | 0 | 2 | 1 | 2 | 6 | 0.24 | 22 | 0.242 |
| X5 | 0 | 0 | 2 | 0 | 1 | 3 | 0.12 | 5 | 0.055 |
|  | | | | | | Σ | 1.00 | 91 | 1.000 |

Розрахунок Рі по 2-й ітерації (рядок результатів перемножується на стовпчик ).

З кожною послідуючою ітерацією значення *Pi*(*k*) уточняються. В процесі вирішення задач багатоцільової оптимізації альтернативні варіанти конкурують між собою і результат оцінювання експерт може подати у вигляді результату «турніра» цих варіантів, тобто у вигляді системи парних порівнянь, як це було показано вище.

Розглянутий метод отриманні експертних оцінок отримав назву метода розстановки приорітетів.

**Код програми:**

<?

namespace App\TPR;

class Lab6 {

public $tprSpreadsheet = array();

public function \_\_construct($data) {

$this->tprSpreadsheet = $data;

}

public function prioritySettingCalculate($data) {

$resultData = array();

$adjacencyMatrix = array();

$sumByRow = array();

$p = array();

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

$adjacencyMatrix[$index] = array();

foreach ($data as $elem) {

if ($column > $elem) {

array\_push($adjacencyMatrix[$index], 2);

} else if ($column < $elem) {

array\_push($adjacencyMatrix[$index], 0);

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

array\_push($adjacencyMatrix[$index], 1);

}

}

}

foreach ($adjacencyMatrix as $row) {

array\_push($sumByRow, array\_sum($row));

}

$p[0] = array();

$sum = array\_sum($sumByRow);

foreach ($sumByRow as $row) {

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

}

$p[1] = array();

foreach ($adjacencyMatrix as $row) {

$s = 0;

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

$s += $column \* $sumByRow[$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);

}

$resultData["adjacencyMatrix"] = $adjacencyMatrix;

$resultData["sumByRow"] = $sumByRow;

$resultData["p"] = $p;

return $resultData;

}

}

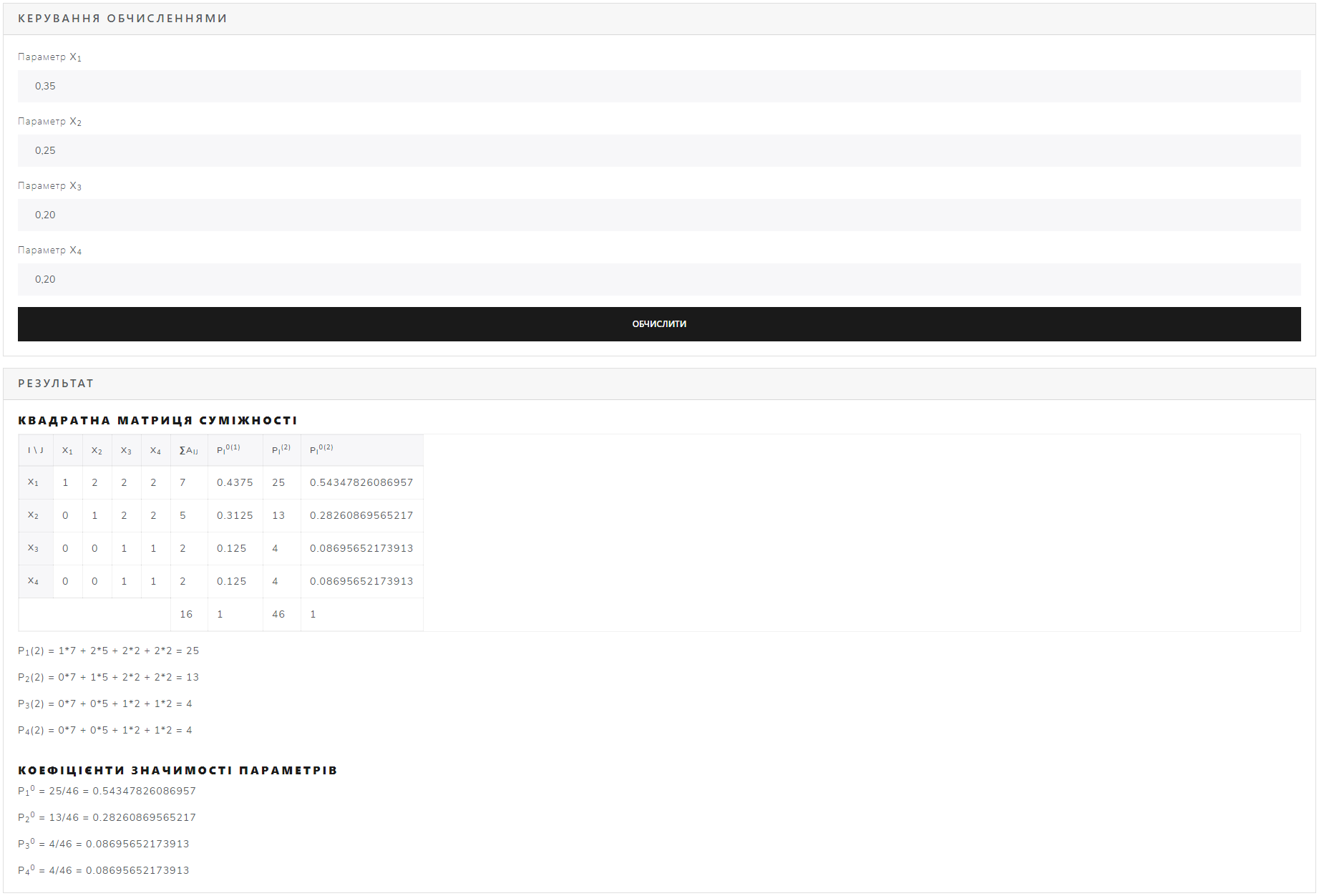
?>

**Результат роботи програми:**

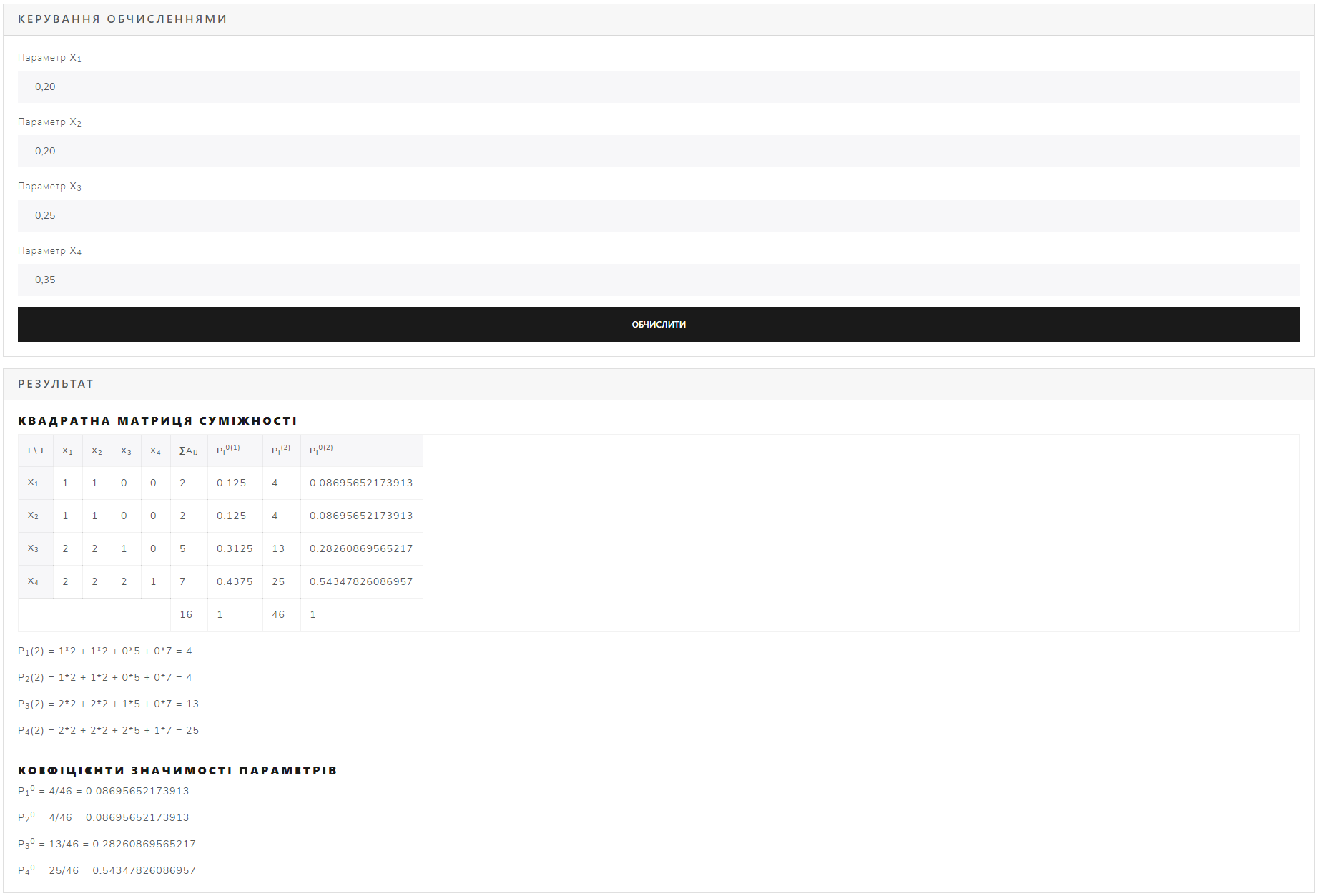
В додаткову панель «Керування обчисленнями», що дозволяє виставляти різні параметри для проведення обрахунків, потрібно ввести значення X1, X2, X3 та X4. Додаткові інструкції з використання даного програмного забезпечення наведені в панелі «Інструкція до керування програмним забезпеченням обраної лабораторної роботи».

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Варіант | Експерт 1 | | | | Експерт 2 | | | | Експерт 3 | | | | Експерт 4 | | | |
|  | I | C | P | N | I | C | P | N | I | C | P | N | I | C | P | N |
| 5 | 0.35 | 0.25 | 0.20 | 0.20 | 0.20 | 0.20 | 0.25 | 0.35 | 0.35 | 0.30 | 0.15 | 0.20 | 0.20 | 0.25 | 0.35 | 0.20 |

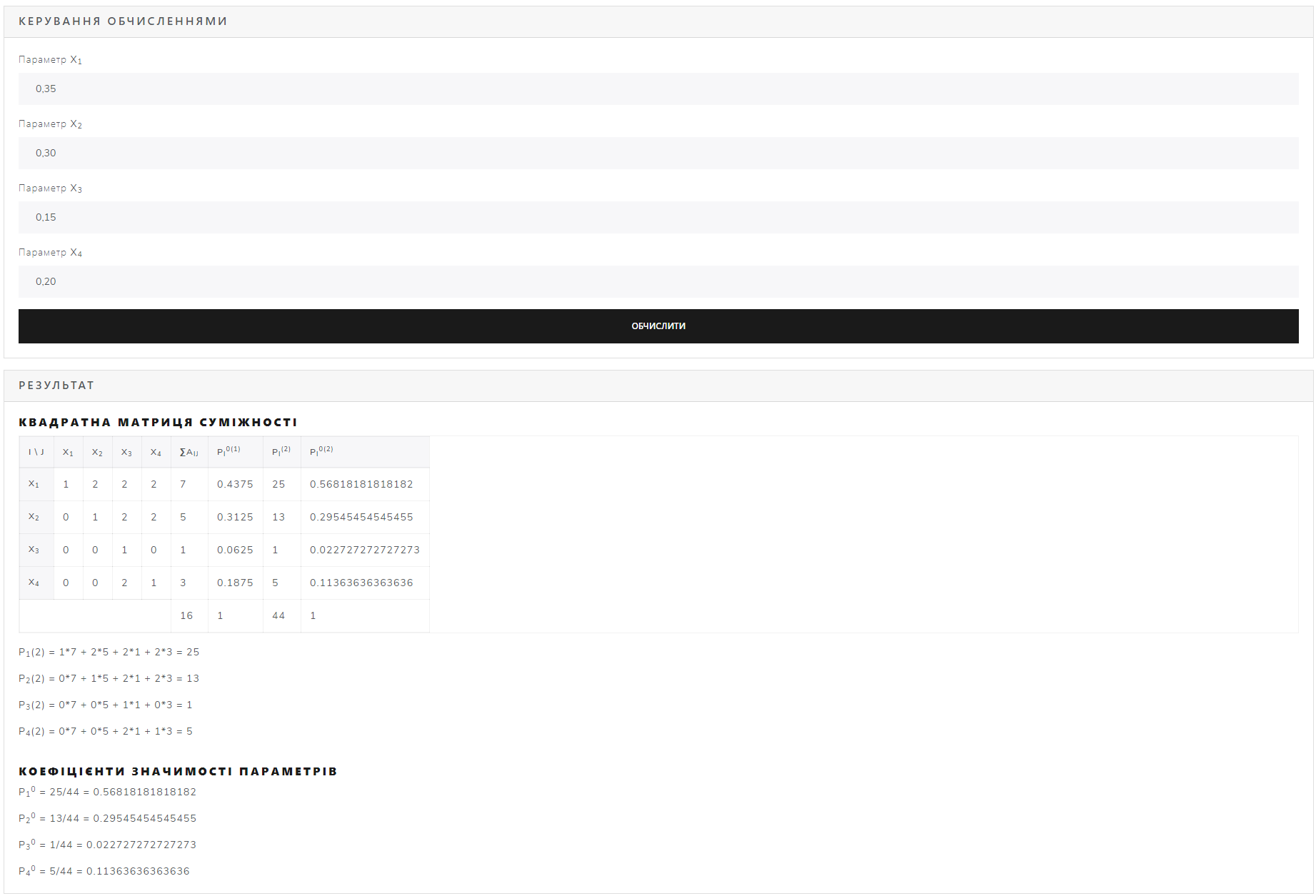
*Таблиця 6.1. Таблиця із завданням для обчислень відповідно до варіанту*



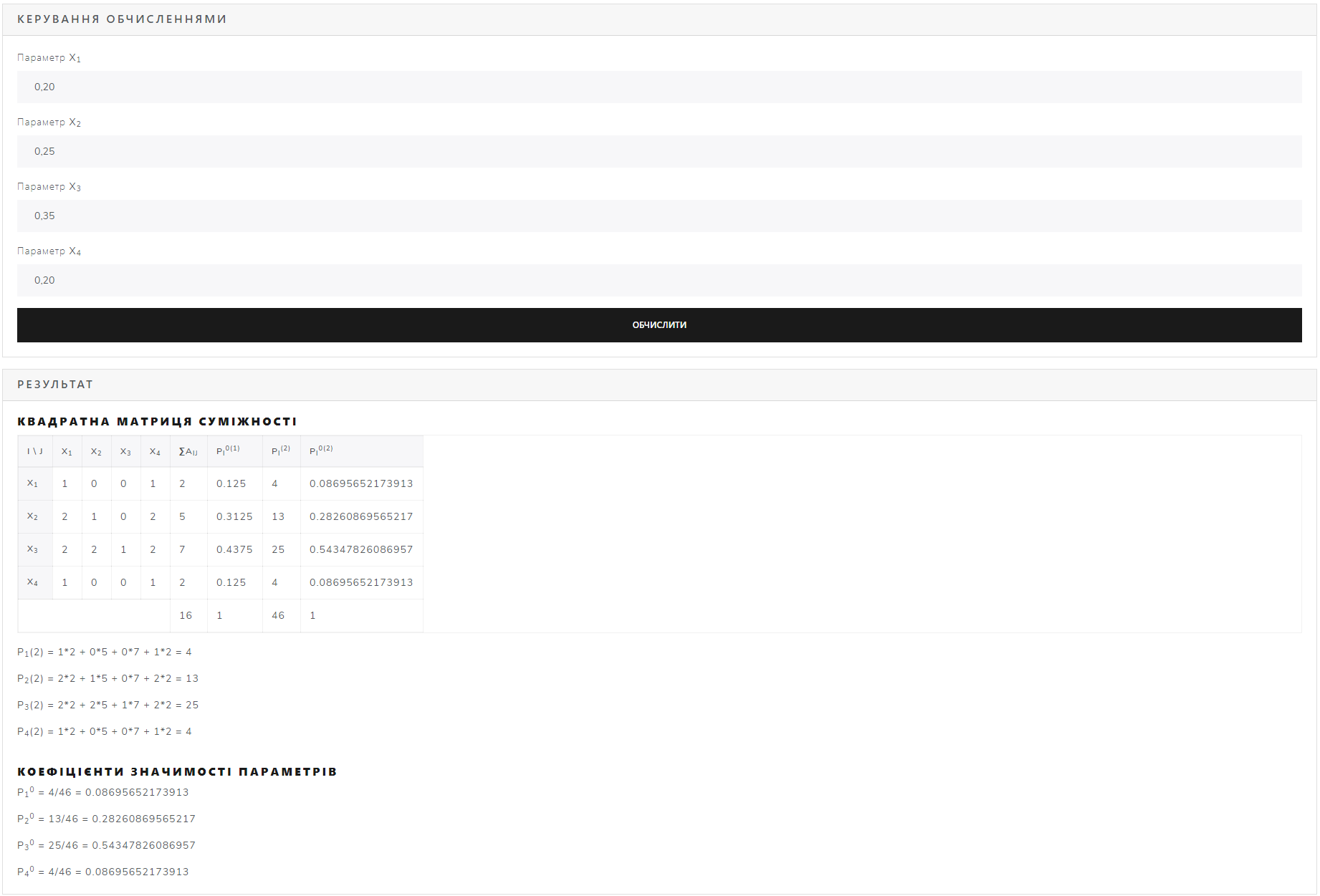
*Рис. 6.1. Визначення коефіцієнтів значимості для оцінок першого експерта*



*Рис. 6.2. Визначення коефіцієнтів значимості для оцінок другого експерта*



*Рис. 6.3. Визначення коефіцієнтів значимості для оцінок третього експерта*



*Рис. 6.4. Визначення коефіцієнтів значимості для оцінок четвертого експерта*

**Висновок**: під час виконання цієї лабораторної роботи я навчився використовувати метод розстановки приорітетів при прийнятті рішень. Закріпив навички використання методу розстановки приорітетів при вирішенні багатоцільових задач прийняття рішень. Створив web-програму для визначення коефіцієнтів значимості параметрів, за допомогою методу розстановки приорітетів. Виконав обчислення коефіцієнтів значимості параметрів в створеній програмі, використовуючи вхідні дані відповідно до варіанту.

# **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 даної 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, ..., *хn*).

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: х1, х2, ... ,хn.

*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.

C1(x1, x2, ..., xn)=0;

C2(x1, x2, ..., xn)=0;

..............................

Cn(x1, x2, ..., xn)=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.

Z1(r1,x1, x2, ..., xn)≤b1;

Z2(r2,x1, x2, ..., xn)≤b2;

. . . . . . . . . . . . . . . . .

Zk(rk,x1, x2, ..., xn)≤bk.

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* найпростіші. 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.

**Результат роботи програми:**

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.

|  |  |  |  |
| --- | --- | --- | --- |
| Матеріали | Норми витрати, г | | Об’єм запасів, г |
| А | Б |
| Мідь | 0.45 | 0.2 | 600 |
| Кадмій | 0.2 | 0.5 | 870 |
| Золото | 0.3 | 0.2 | 918 |
| Прибуток від реалізації, грн/10 000 | 580 | 290 |

*Таблиця 7.1. Таблиця із значеннями коефіцієнтів відповідно до варіанту*

*Цільова функція:* F = 580x1 + 290x2 → max, при системі обмежень:

0.45x1 + 0.2x2 ≤ 600,

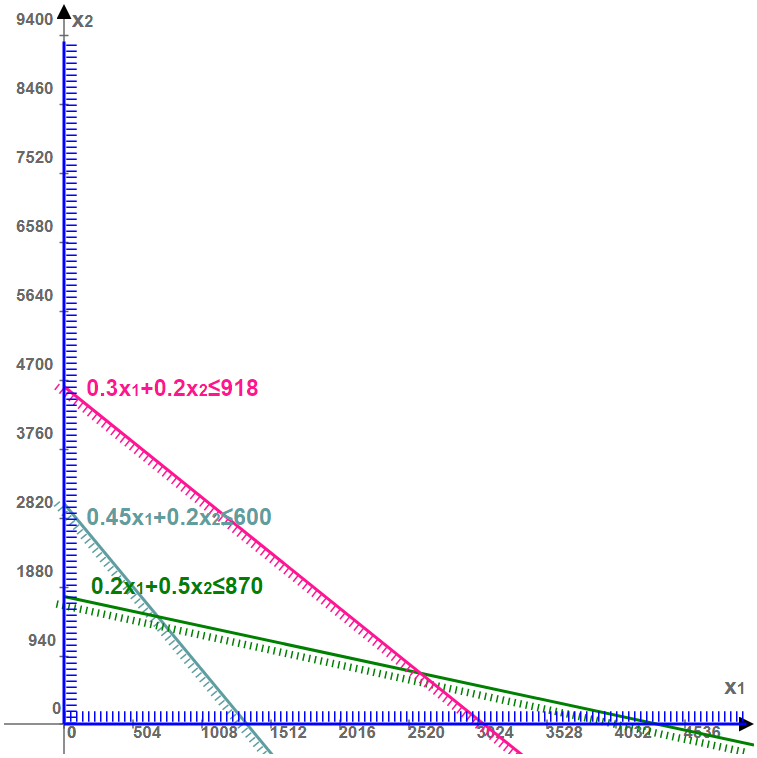
0.2x1 + 0.5x2 ≤ 870,

0.3x1 + 0.2x2 ≤ 918,

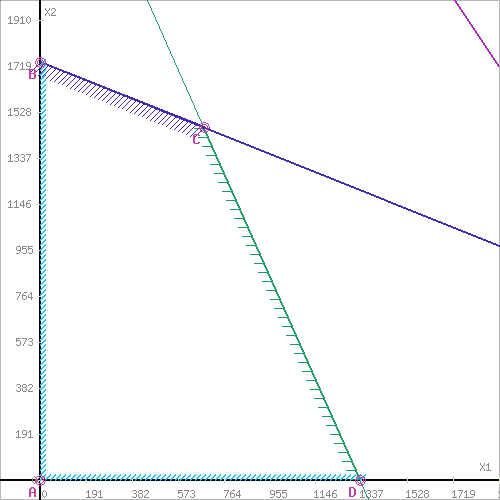
x1 ≥ 0,

x2 ≥ 0.

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



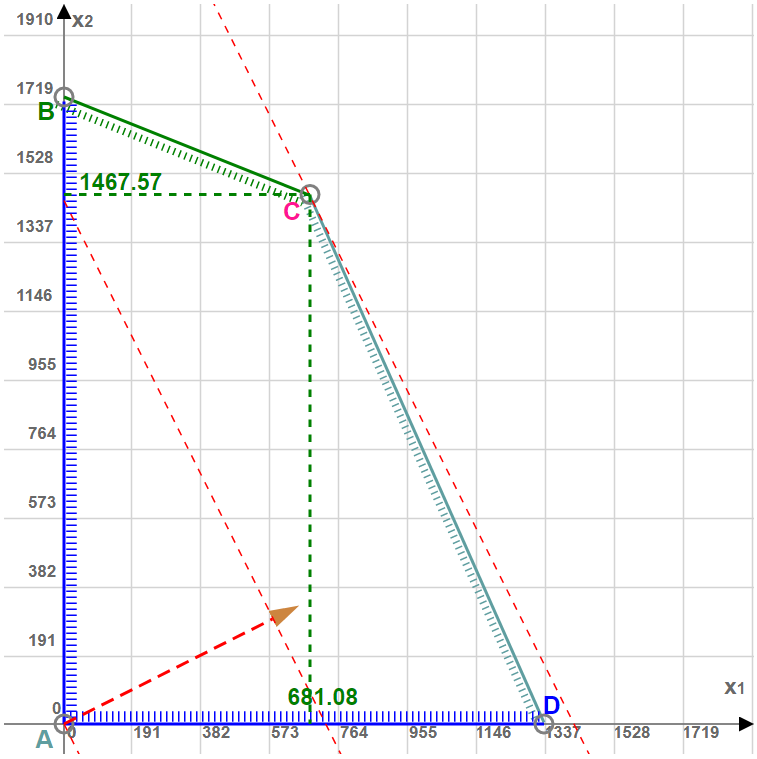
Перетином напівплощин буде область, координати точок якого задовольняють умові нерівностей системи обмежень задачі.



Побудуємо пряму, що відповідає значенню функції:

F = 580x1 + 290x2.

Вектор-градієнт, складений з коефіцієнтів цільової функції, вказує напрямок максимізації F(x). Початок вектора - точка (0; 0), кінець - точка (580; 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:

0.45x1 + 0.2x2 = 600,

0.2x1 + 0.5x2 = 870.

Вирішивши систему рівнянь, отримаємо:

x1 = 681.0811,

x2 = 1467.5676.

Максимальне значення цільової функції:

F (X) = 580 × 681.0811 + 290 × 1467.5676 = 820621.642‬.

Максимальний прибуток становить: 82.06 грн.

**Висновок**: під час виконання цієї лабораторної роботи я ознайомився із графічним методом вирішення задач оптимізації. Закріпив навички використання графічного методу при вирішенні оптимізаційних задач. Використовуючи графічний метод вирішення задач оптимізації, обчислив максимізований прибуток при обмежених ресурсах за умовами завдання відповідно до варіанту.