МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТРАНСПОРТНИЙ УНІВЕРСИТЕТ

КАФЕДРА «МІЖНАРОДНІ ПЕРЕВЕЗЕННЯ ТА МИТНИЙ КОНТРОЛЬ»

**КУРСОВА РОБОТА**

**з навчальної дисципліни «АЛГОРИТМІЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ОПТИМІЗАЦІЇ ВАНТАЖОПОТОКІВ В ТРАНСПОРТНИХ СИСТЕМАХ»**

**на тему**

«РОЗРОБКА ПРОГРАМНОГО КОМПЛЕКСУ ЗНАХОДЖЕННЯ ОПТИМАЛЬНОГО ПЛАНУ ПЕРЕВЕЗЕНЬ В ТРАНСПОРТНІЙ МЕРЕЖІ»

Виконав: студент групи КН-V-1 Маташин М.С.

Перевірив: проф.Прокудін Г.С.

Київ 2014

# **Зміст**

[**Зміст** 2](#_Toc405923228)

[**Вступ** 4](#_Toc405923229)

[**Розділ 1. Постановка задачі** 5](#_Toc405923230)

[**Розділ 2. Теоретичні відомості** 7](#_Toc405923233)

[2.1. Метод південно– західного кута 7](#_Toc405923234)

[2.2. Модифікований метод Дейкстри 11](#_Toc405923235)

[2.3. Метод потенціалів 13](#_Toc405923236)

[**Розділ 3. Опис роботи створеного програмного продукту та програмний код** 13](#_Toc405923237)

[3.1. Опис роботи та можливостей програми 13](#_Toc405923238)

[3.2. Програмний код 17](#_Toc405923239)

[**Висновок** 44](#_Toc405923240)

[**Список використаної літератури** 45](#_Toc405923241)

# **Вступ**

Сучасні міжнародні умови, до яких прагне Україна, вимагають в галузі логістики вантажних перевезень усе більшої уваги, стрімкого зростання та вдосконалення. Ефективність та якість вантажних перевезень значно залежать від оптимізації процесів координації роботи різних видів транспорту, раціонального розподілу між ними обсягів перевезень, своєчасного формування необхідних управлінських рішень. Найперше, особливу увагу при цьому потрібно звернути на два найважливіших показники транспортного обслуговування – вартість здійснення транспортних перевезень та строки виконання замовлень на доставку вантажів.

Аналіз наявних у вітчизняній і світовій практиці підходів дооптимізації перевезень пасажирів і вантажів у ТС виявив низку недоліків:

* неспроможність планувати перевезення пасажирів і вантажів із довільно орієнтованою матрицею транспортних кореспонденцій;
* наявність істотних обмежень на розмірність розв'язуваних транспортних завдань;
* недостатнє використання в перевізному процесі технологій спільної взаємодії різних видів транспорту;
* неможливість чіткої математичної формалізації більшості методів оптимізації перевезень на ТМ, що у свою чергу приводить до неможливості використання сучасних засобів інформаційних технологій.

Більшу частини перерахованих вище недоліків по оптимальному плануванню і маршрутизації вантажних перевезень у транспортних системах знято за допомогою використання *матрично-мережевої моделі* представлення й управління вантажними перевезеннями на транспортних мережах.

# **Розділ 1. Постановка задачі**

Виконання курсової роботи має на меті закріплення практичних навичок у вирішенні практичних завдань по оптимізації вантажних перевезень в транспортних системах (ТС).

Курсова робота припускає розробку програмного комплексу, який працює з побудованою *матрично-мережевою моделлю* (*МММ*) здійснення вантажних перевезень в транспортних системах. Робота з *МММ* включає наступні етапи:

* створення матриці транспортних кореспонденцій (матриці суміжності) для відповідної транспортної мережі (ТМ);
* зведення за допомогою модифікованого методу Дейкстри сітьового представлення перевезень вантажу на ТМ до табличного виду – транспортної таблиці (ТТ);
* побудову у ТТ одним з відомих методів опорного плану перевезень вантажу;
* перевірку правильності побудованого опорного плану методом прокрутки;
* побудову маршрутів транспортування вантажів у відповідності з оптимальним планом перевезень вантажу на ТМ.

Проектування програмного комплексу ведеться у відповідності з наступними технологічними принципами: покрокової деталізації; структурного та об’єктно-орієнтованого програмування; використанням різних абстрактних типів даних.

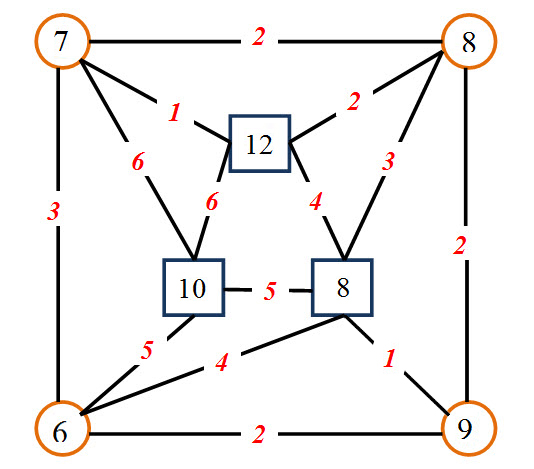


Рис.1. Граф варіанту №15

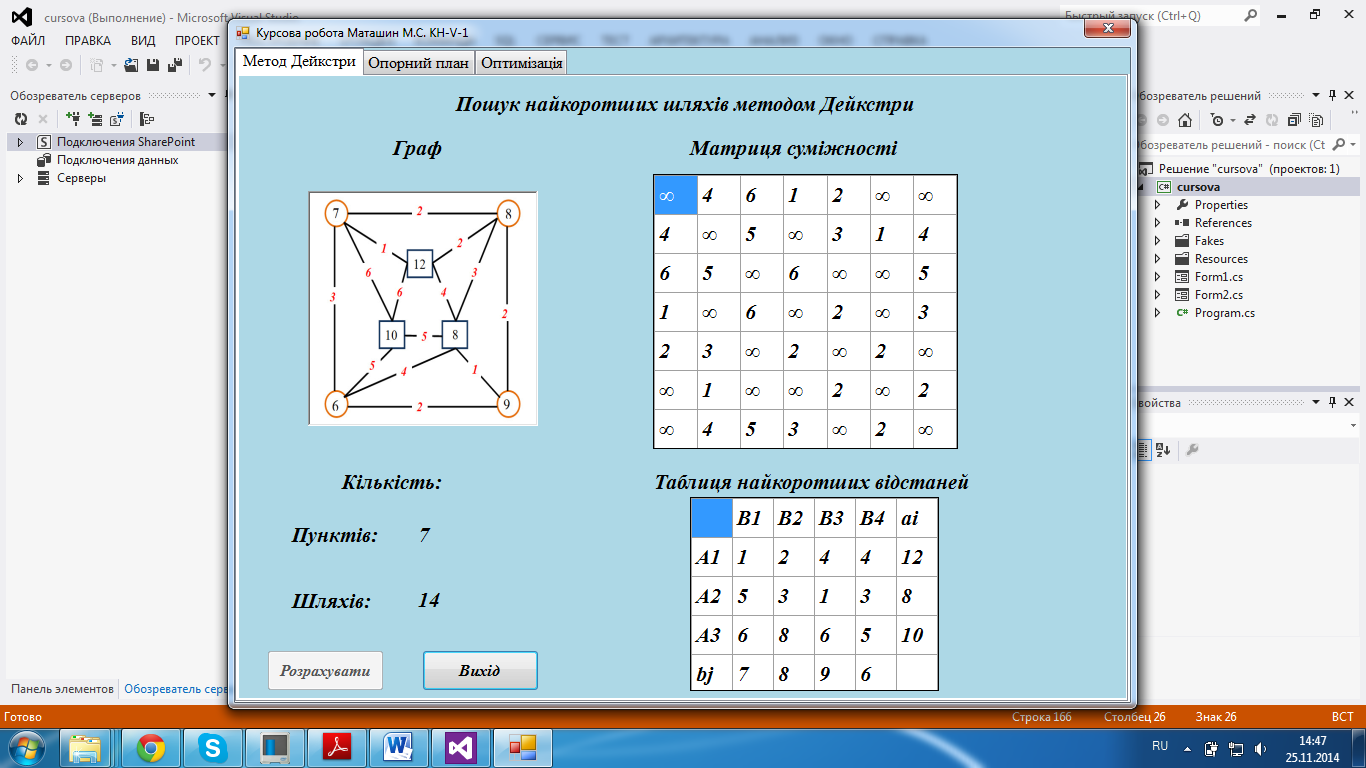
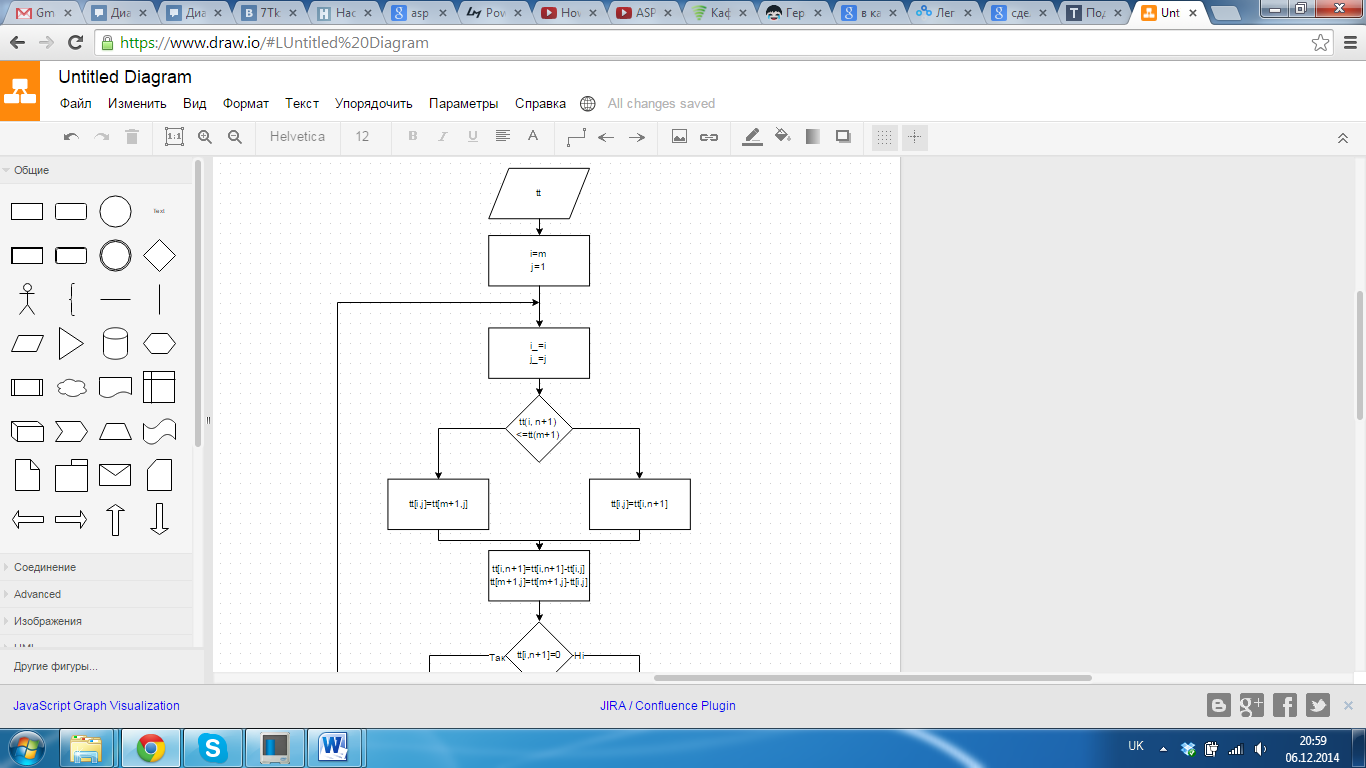
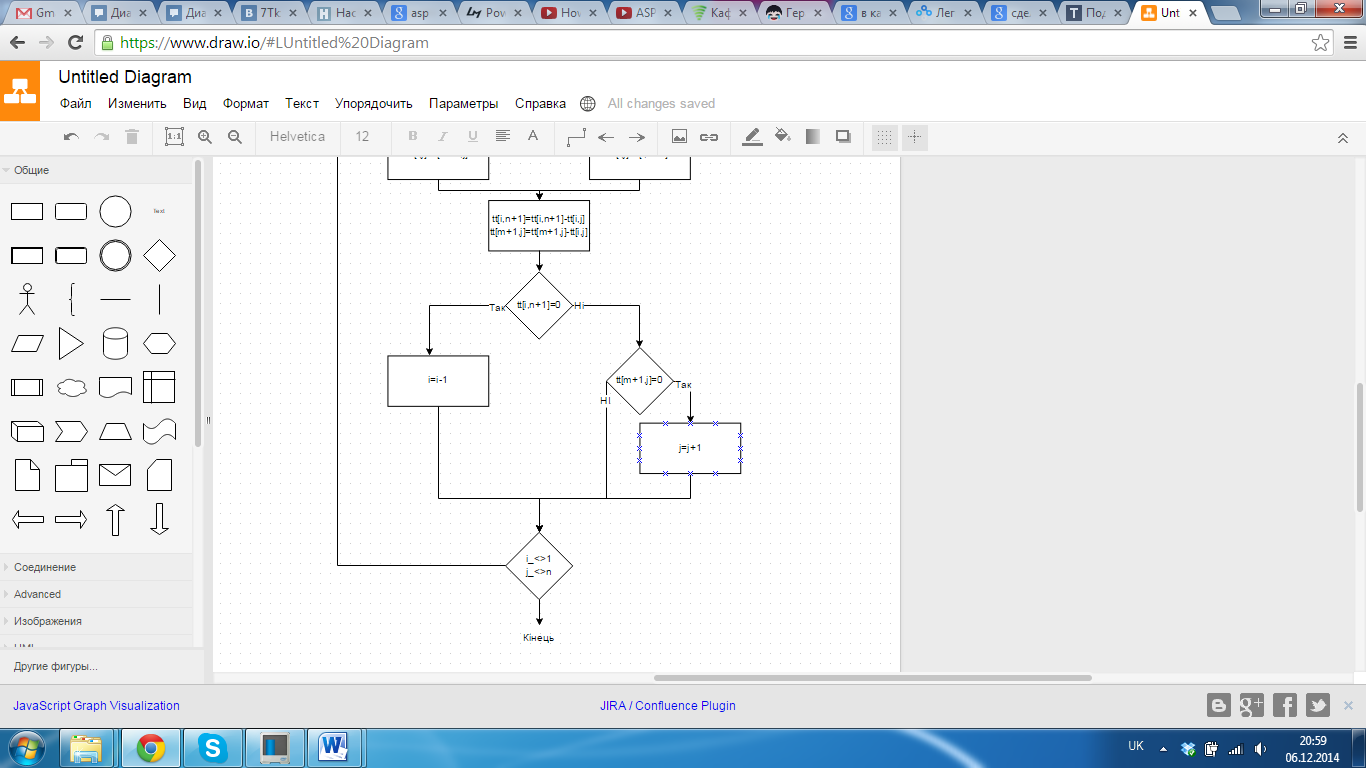


Рис.2. Матриця суміжності

# **Розділ 2. Теоретичні відомості**

## 2.1. Метод південно– західного кута





Блок-схема методу методу південно-західного кута

Процес пошуку опорного плану перевезень за методом південно-західного кута розглянемо на прикладі табл.1:

*Таблиця*1

ВихіднаТТ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **B1** | **B2** | **B3** | **B4** | ***Запаси***  *ai* |
| **A1** | **1**  *x11* | **2**  *x12* | **4**  *x13* | **4**  *x14* | **12** |
| **A2** | **5**  *x21* | **3**  *x22* | **1**  *x23* | **3**  *x24* | **8** |
| **A3** | **6**  *x31* | **8**  *x32* | **6**  *x33* | **5**  *x34* | **10** |
| ***Заявки***  *bj* | **7** | **8** | **9** | **6** | **30** |

Заповнення обсягів перевезень починаємо з самої нижньої лівої клітинки А3B1 (звідси назва методу). Принцип заповнення: задовольнити максимально можливий обсяг замовлення b1; якщо обсягу a3 не вистачає, беремо частину від , якщо в  щось залишається, віддаємо решту a3 до b2 і т.д. Для розглянутого випадку робимо наступне:

1. Задовольнимо b1 за рахунок A3, решту a3=(10-7)=3 відправимо до *B2* (див. табл. 2).
2. Оскільки *В2* ще не задовольнили, додамо необхідний обсяг за рахунок *А2* (ще 5); решту a2=(8-5=3) відправимо до *В3* (див. табл. 3);
3. Щоб повністю задовольнити *В3*, додамо необхідний обсяг за рахунок *A1* (ще 6). Решту а1=(12-6)=6 відправимо до *В4*, задовольнив таким чином його повністю (див. табл. 4).

*Таблиця*2

ТТ з розподілом запасів А3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **B1** | **B2** | **B3** | **B4** | ***Запаси***  *ai* | ai' |
| **A1** | **1** | **2** | **4** | **4** | 12 | **12-6=6** |
| **A2** | **5** | **3** | **1** | **3** | **8** | **8-5=3**  **3-3=0** |
| **A3** | **6**  **7** | **8**  3 | **6** | **5** | **10** | **10-7=3**  **3-3=0** |
| ***Заявки***  *bj* | **7** | **8** | **9** | **6** | **30** |  |
| *bj'* | **7-7=0** | **8-3=5** |  | **6-6=0** |  |  |

*Таблиця*3

ТТ з розподілом запасів А2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **B1** | **B2** | **B3** | **B4** | ***Запаси***  *ai* | ai' |
| **A1** | **1** | **2** | **4** | **4** | 12 |  |
| **A2** | **5** | **3**  5 | **1**  3 | **3** | **8** | **8-5=3**  **3-3=0** |
| **A3** | **6** | **8** | **6** | **5** | **10** |  |
| ***Заявки***  *bj* | **7** | **8** | **9** | **6** | **30** |  |
| *bj'* |  | **8-3=5**  **5-5=0** | **9-3=6** |  |  |  |

Підрахуємо кількість ненульових перевезень (зайнятих клітинок). Таких є 6, що відповідає обов'язкової умові (*m + n – 1*) = (3 + 4 – 1) = 6. Оскільки суми перевезень по рядах і колонках відповідають  і , отриманий план перевезень є можливим і опорним, тому що цей план є початковим. Для отриманого плану можливо підрахувати загальні витрати на здійснення всіх перевезень, тобто

L0=7\*6+3\*8+5\*3+3\*1+6\*4+6\*4=132 *у.г.о.*

У розділі 3 представлений текст процедури на алгоритмічній мові C# побудови опорного плану перевезень вантажу методом південно-західного кута.

*Таблиця*4

ТТ з розподілом запасів

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **B1** | **B2** | **B3** | **B4** | ***Запаси***  *ai* | ai' |
| **A1** | **1** | **2** | **4**  6 | **4**  **6** | 12 | **12-6=6**  **6-6=0** |
| **A2** | **5** | **3**  5 | **1**  3 | **3** | **8** | **8-5=3**  **3-3=0** |
| **A3** | **6**  **7** | **8**  3 | **6** | **5** | **10** | **10-7=3**  **3-3=0** |
| ***Заявки***  *bj* | **7** | **8** | **9** | **6** | **30** |  |
| *bj'* | **7-7=0** | **8-3=5**  **5-5=0** | **9-3=6**  **6-6=0** | **6-6=0** |  |  |

## 2.2. Модифікований метод Дейкстри

У якості вихідних даних для цієї задачі беремо граф, зображений на рис. 1, що містить час польоту по маршрутах, що зв'язує визначені міста, причому вся множина його вершин V розбита на дві підмножини. До пер­шої підмножини відносяться пункти запасів ( m міст ), а друга підмножина містить пункти заявок ( п міст ). Потрібно відшукати найкоротші маршрути доставок між пунктами запасів і заявок, включаючи пересадки в проміжних пунктах (ними можуть бути як пункти запасів, так і пункти заявок). Іншими словами ми повинні одержати матрицю найкоротших відстаней пунктами запасів і пунктами заявок.

Для рішення цієї задачі алгоритми, наприклад, Дейкстри або Флойда, не застосовні. Алгоритм Дейкстри недостатній для її рішення( він знаходить усього лише один рядок з матриці найкоротших відстаней ), а алгоритм Флойда надлишковий(він генерує матрицю найкоротших відстаней між будь-якими пунктами).

**Алгоритм модифікованого методу Дейкстри**

procedureNew( var D: array [ I .. m, 1 .. (m + n)] of real;

C: array[l .. (m + n), 1 ..(m + n)] of real;

P: array[l .. m, 1.. (m + n)] of integer);

begin

(1) for і := 1 to m do

begin

S := {і}; { вибір чергової вершини з підмножини аеропортів вильотів }

for j := 1 to (m + n) do

begin

D[i, j] := C[i, j]; { ініціалізація D }

P[i,j] :=i end

(2)for j := 1 to (m + n-1) do

begin

вибір з множини V\Sтакої вершини w, що значення D[i,w]

мінімальне;

додати wдо множини S;

forкожна вершина vз множини V\Sdo

begin

if (D[i, w) + C[w, v] <D[i, v] then

P[i,j] :=w;

D[i, v] := min(D[i, v], D([i, w] + C[w, v] );

end

(3) end

(4) end  
end; { New }

Тут масив *D* являє собою результуючу матрицю найкоротших відстаней, причому на кожному кроці елемент *D[i,* v]містить довжину поточного найкоротшого шляху від вершини *i* до вершини *v*. Масив *С* задає вартості перельотів, де елемент С[і, j]дорівнює вартості дуги і—>*j.* Якщо дуги *i* —>*j*не існує, то С[і, j]є рівним ∞(нескінченності), тобто більшим будь-якої фактичної вартості дуг. Елемент масиву Р[і, v]містить номер вершини, безпосередньо попередній вершині *v*у найкоротшому шляху від вершини і.Множина *S* має той же зміст, що й в алгоритмі Дейкстри.

Зовнішній цикл( рядка 1 - 4 ) здійснює послідовний перебір всіх пунктів запасів, а внутрішній цикл( рядка 2 - 3 ) знаходить найкоротші маршрути від цих пунктів до всіх інших, причому якщо в цьому маршруті присутні проміжні ве­ршини, то вони запам'ятовуються.

Застосуємо алгоритм модифікованого методу Дейкстри для орієнтованого графа, зображеного на 1. В якості пунктів запасів використаємо вершини А1,А2,А3, а в якості пунктів заявок – вершини В1,В2,В3, В4. В результаті отримаємо матрицю найкоротших відстаней між цими пунктами:

## 2.3. Метод потенціалів

У методі потенціалів кожному рядку i і кожному стовпцю j транспортної таблиці ставляться у відповідність числа (потенціали) ui і vj. Для кожної базисної змінної xij, потенціали ui і vjзадовольняють рівнянню:

ui + vj= cij,

Щоб знайти значення потенціалів з цієї системи рівнянь, потрібно присвоїти одному з них довільне значення (зазвичай вважають u1 = 0) і потім послідовно обчислювати значення інших потенціалів.

Далі, використовуючи знайдені значення потенціалів, для кожної небазисной змінної обчислюються величини

ui + vj= cij.

Якщо всі ці числа є недодатними то опорний план є оптимальним і розв'язування на цьому завершується. В іншому випадку знаходиться найбільше додатне значення і відповідна йому змінна вводиться в базис. Для визначення змінної, що виводиться з базису будується послідовність:

xij -> xi1j1 -> xi2j2 ->…-> xij,

де xij — змінна, що вводиться в базис, а всі інші змінні є базисними. Окрім цього в цій послідовності при переході на кожному етапі одна координата залишається незмінною і якщо при певному переході незмінною була перша координата, то на наступному незмінною буде друга. Якщо зображувати перехід між змінними на транспортній таблиці стрілками між відповідними клітинами це оначає, що переходи можуть бути лише вертикальними чи горизонтальними, але не діагональними, і також після горизонтального переходу має йти вертикальний і навпаки.

Після побудови послідовності

xij -> xi1j1 -> xi2j2 ->…-> xij,

можна записати значення відповідних змінних і знайти мінімальне значення серед чисел, що стоять на непарних позиціях. Наступним кроком це число слід додати до всіх змінних, що стоять на парних позиціях і відняти від всіх змінних, що стоять на непарних. Змінна якій відповідало найменше число виводиться з базиса.

В такий спосіб одержується новий опорний план і до нього можна знову застосувати ті ж дії.

*Таблиця* 5

ТТ з потенціалами та контуром перерозподілу

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **B1** | **B2** | **B3** | **B4** |  | ***Запаси***  *ai* |
| **A1** | **1** | **+ 2** | **- 4**  **6** | **4**  **6** | **0** | **12** |
| **A2** | **5** | * **3**   **5** | **+ 1**  ***3*** | **3** | **-3** | **8** |
| **A3** | **6**  ***7*** | **8**  ***3*** | **6** | **5** | **2** | **10** |
|  | **4** | **6** | **4** | **4** |  |  |
| ***Заявки***  *bj* | **7** | **8** | **9** | **6** |  | **30** |

А1В1: 4+0>1 (3) -

**A1B2: 6+0>2 (4) -**

A2B1: 4-3<5 +

A2B4: 4-3<3 +

A3B3: 4+2=6 +

A3B4: 4+2> 5 (1) –

В такий спосіб одержується новий опорний план і до нього можна знову застосувати ті ж дії.

*Таблиця* 6

ТТ з потенціалами та контуром перерозподілу

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **B1** | **B2** | **B3** | **B4** |  | ***Запаси***  *ai* |
| **A1** | **1**  **4** | **2**  **8** | **4** | **4**  **6** | **0** | **12** |
| **A2** | **5** | **3** | **1**  ***8*** | **3** | **0** | **8** |
| **A3** | **6** | **8** | **6**  **1** | **5**  **6** | **0** | **10** |
|  | **1** | **2** | **1** | **0** |  |  |
| ***Заявки***  *bj* | **7** | **8** | **9** | **6** |  | **30** |

А1В3: 1+0=1<4 +

A1B4: 0+0=0<4 +

A2B1: 1+0=1<5 +

A2B4: 0+0=0<3 +

A3B2: 2+5=7<8 +

Виконуються всі умови тож цей план опорний план є оптимальний.

Для отриманого плану підрахуємо загальні витрати на здійснення всіх перевезень:

L=4\*1+8\*2+8\*1+3\*6+1\*6+6\*5=82 *у.г.о.*

*L0=132<L=82*

Ці витрати менші від початкових, тому оптимізація методом потенціалів працює правильно.

# **Розділ 3. Опис роботи створеного програмного продукту та програмний код**

## 3.1. Опис роботи та можливостей програми

Після запуску програми відкривається головне вікно:

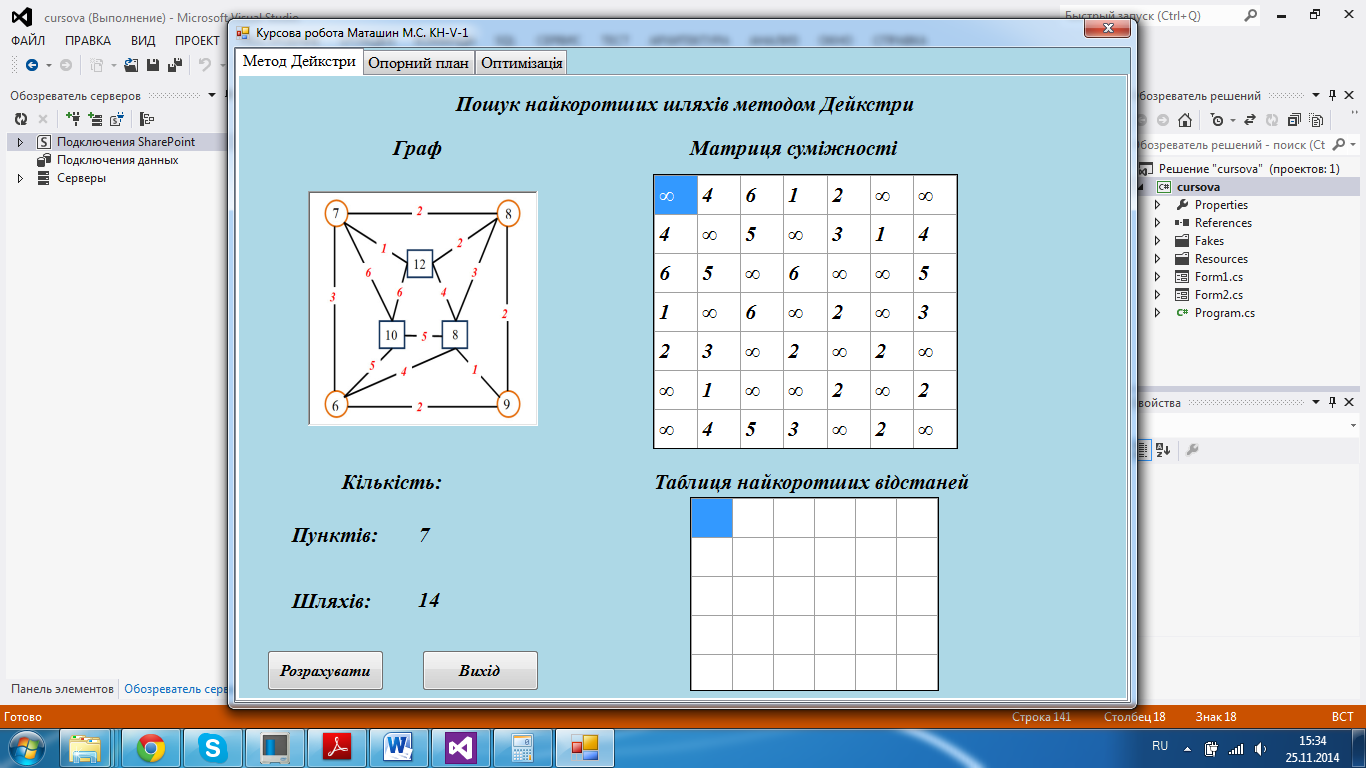


Рис.3. Головне вікно програми

Натискаємо кнопку «Розрахувати» знаходить та виводить на екран матрицю найкоротших відстаней між пунктами запасів та пунктами заявок. На екрані з’явиться результат роботи. Також в вкладці «Опорний план» стає доступною кнопка «Побудувати»:

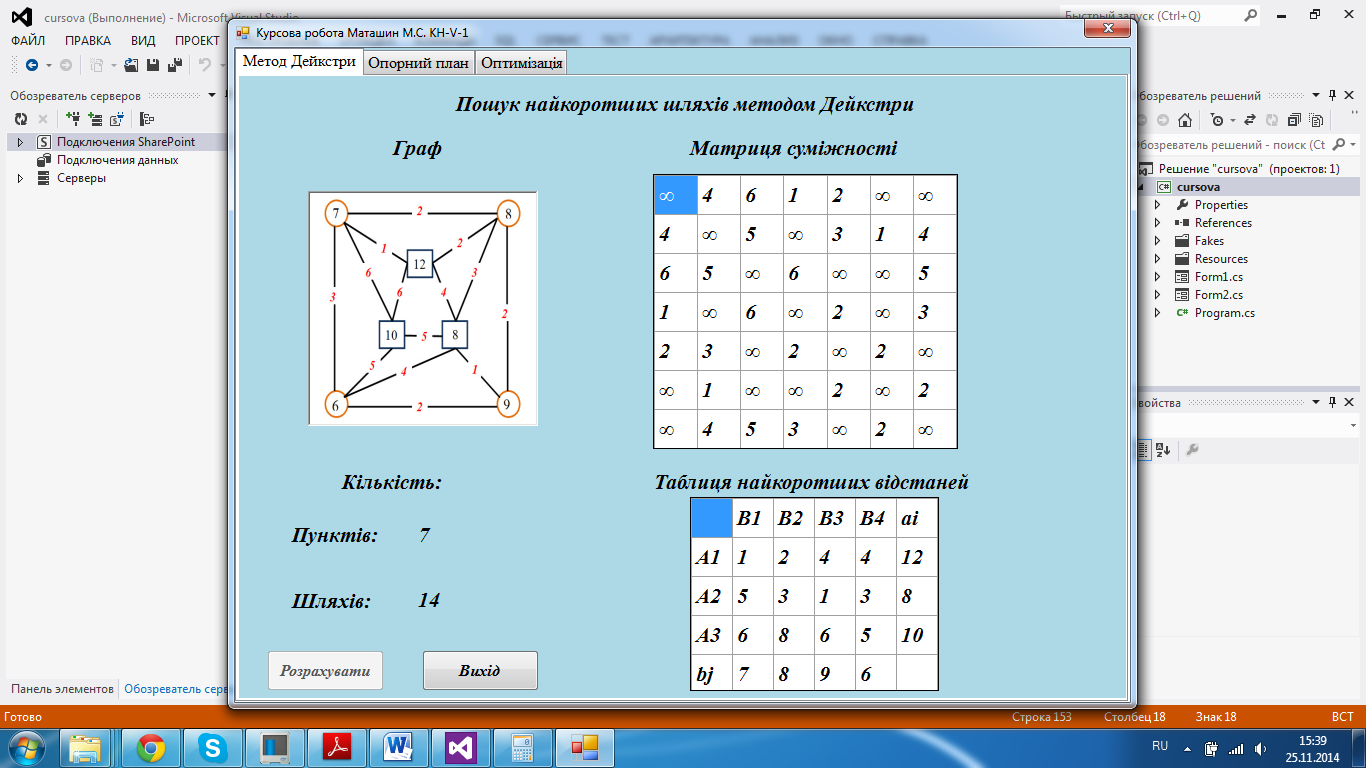


Рис.4. Пошук найкоротших шляхів

Кнопка «Побудувати» будує та виводить на екран опорний план маршрутів перевезень за допомогою методу південно-західного кута. Також на екран виводяться маршрути перевезень з кількостями вантажів, що перевозяться по цим маршрутам, сума вартості всіх перевезень(в умовних грошових одиницях), а також перевірка правильності побудованого опорного плану за допомогою методу прокрутки(виводиться в окрему таблицю):

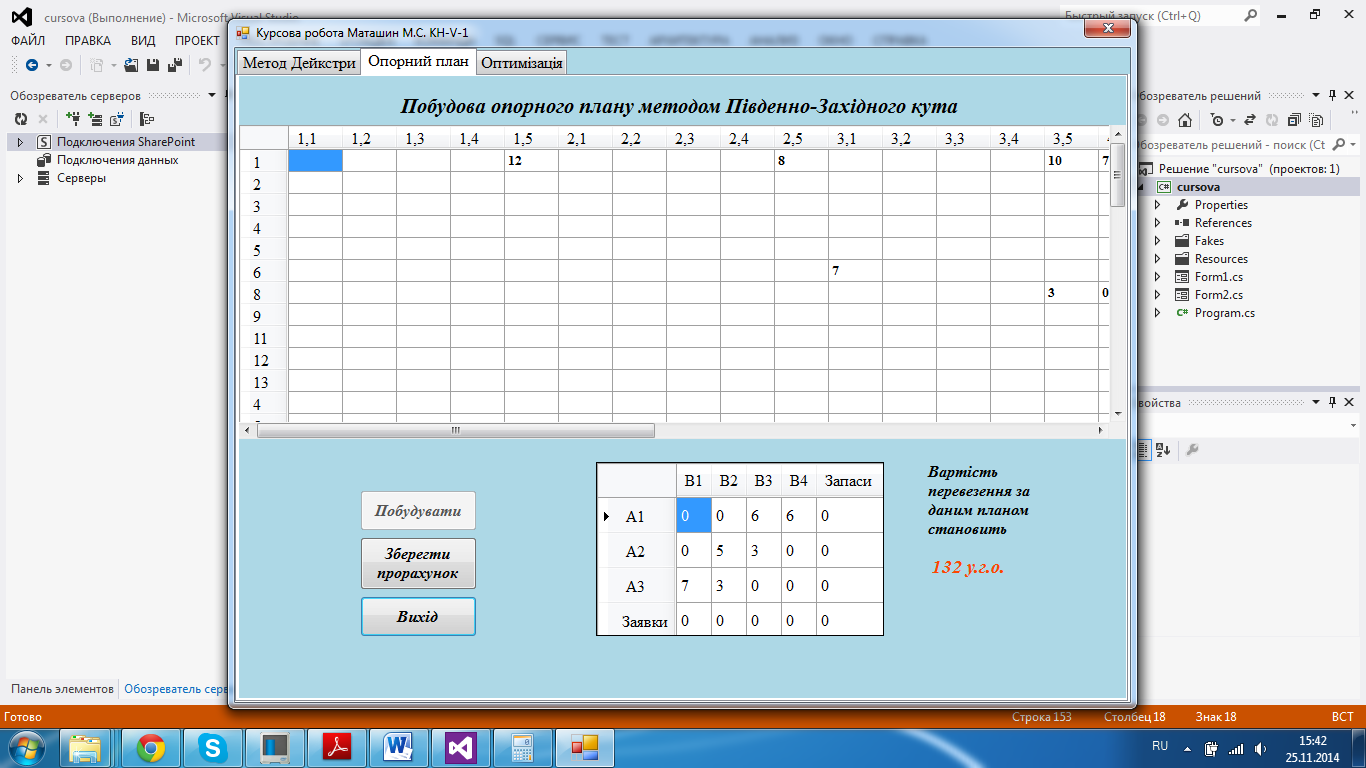


Рис.5. Побудова опорного плану

На вкладці «Оптимізація» натискаємо кнопку «Оптимізувати» знаходимо оптимальний план по методу потенціалів оптимізації транспортних перевезень та виводимо таблицю потенціалів на екран. Також бачимо вартість перевезення за оптимальним планом. В полі «Результат» бачимо результат роботи всіх обчислень:

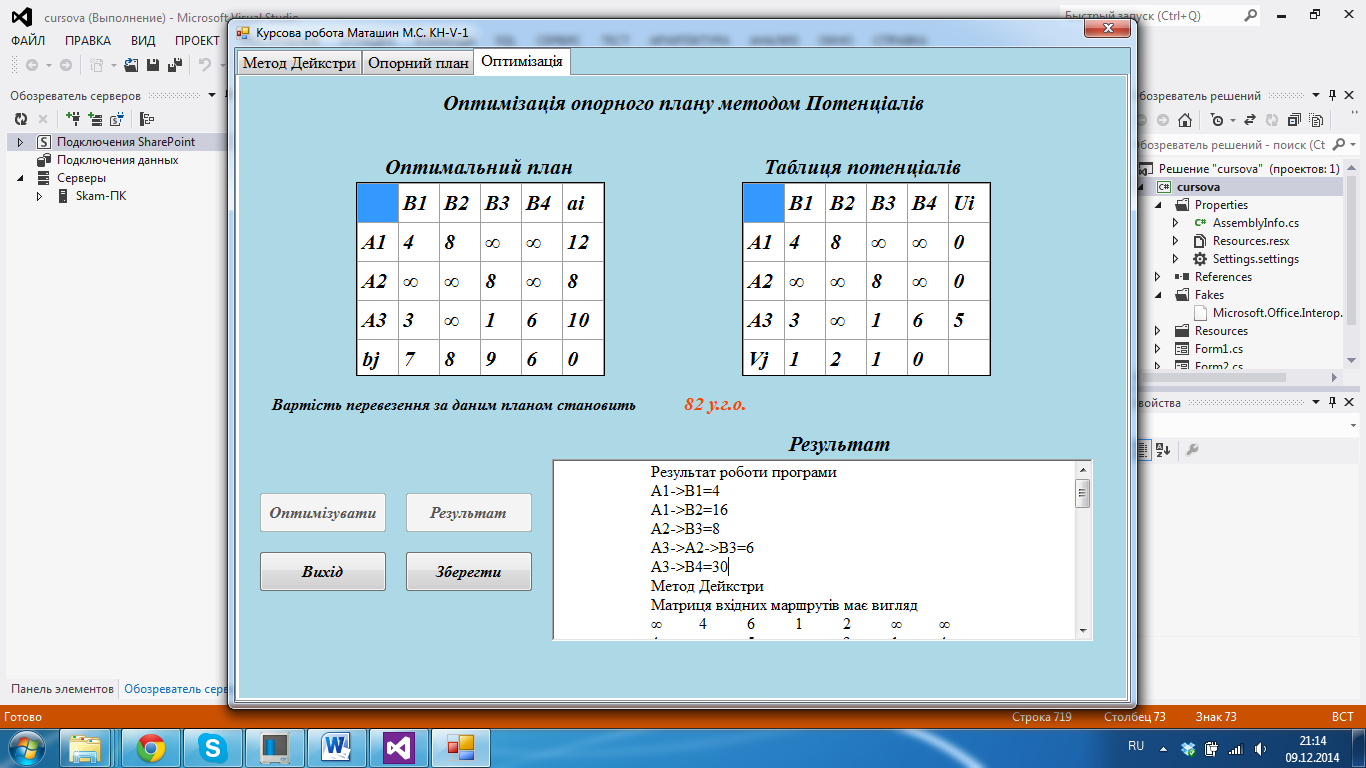


Рис.6. Оптимізація опорного плану методу Потенціалів

## 3.2. Програмний код

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

using System.IO;

namespace cursova

{

public partial class FormMain : Form

{

// Initialization of variables

static Int32 inf = 2147000000;

int[,] ribs = {{inf,4,6,1,2,inf,inf},

{4,inf,5,inf,3,1,4},

{6,5,inf,6,inf,inf,5},

{1,inf,6,inf,2,inf,3},

{2,3,inf,2,inf,2,inf},

{inf,1,inf,inf,2,inf,2},

{inf,4,5,3,inf,2,inf}};

int[,] TT = {

{0,0,0,0,12},

{0,0,0,0,8},

{0,0,0,0,10},

{7,8,9,6,0}

};

int[,] d2 = new int[3, 4];

int[,] p2 = new int[7, 7];

//Initialization of Form

public FormMain()

{

InitializeComponent();

}

private void FormMain\_Resize(object sender, EventArgs e)

{

tabCntrl.Left = (FormMain.ActiveForm.Width - tabCntrl.Width) >> 1;

tabCntrl.Top = (FormMain.ActiveForm.Height - tabCntrl.Height) >> 1;

}

private void FormMain\_Load(object sender, EventArgs e)

{

DoubleBuffered = true;

labelDH.Left = (tabCntrl.Width - labelDH.Width) >> 1;

dataGridViewDM.ColumnCount = ribs.GetLength(1);

dataGridViewDM.RowCount = ribs.GetLength(0);

TNVDG.ColumnCount = 6;

TNVDG.RowCount = 5;

OPDG.ColumnCount = 6;

OPDG.RowCount = 5;

PODG.ColumnCount = 6;

PODG.RowCount = 5;

labelDVR.Text = (ribs.GetLength(1)).ToString();

int k=0;

for (int i = 0; i < ribs.GetLength(0); i++)

for (int j = 0; j < ribs.GetLength(1); j++)

if (ribs[i, j] != inf)

k++;

labelDRR.Text =(k/2).ToString();

for (int i = 0; i < ribs.GetLength(0); i++)

for (int j = 0; j < ribs.GetLength(1); j++)

if (ribs[i, j] != inf)

{

dataGridViewDM.Rows[i].Cells[j].Value = ribs[i, j];

}

else

{

dataGridViewDM.Rows[i].Cells[j].Value = "∞";

}

//////////////////////////

calc.RowHeadersDefaultCellStyle.Padding = new Padding(3);

calc.RowHeadersDefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleLeft;

TTG.RowHeadersDefaultCellStyle.Alignment = DataGridViewContentAlignment.MiddleCenter;

TTG.RowHeadersWidth = 80;

TTG.RowCount = 4;

TTG.Rows[0].HeaderCell.Value = ("А1").ToString();

TTG.Rows[1].HeaderCell.Value = ("А2").ToString();

TTG.Rows[2].HeaderCell.Value = ("А3").ToString();

TTG.Rows[3].HeaderCell.Value = ("Заявки").ToString();

for (int i1 = 0; i1 < TT.GetLength(0); i1++)

for (int j1 = 0; j1 < TT.GetLength(1); j1++)

TTG.Rows[i1].Cells[j1].Value = TT[i1, j1];

}

private void tabCntrl\_Resize(object sender, EventArgs e)

{

labelDH.Left = (tabCntrl.Width - labelDH.Width) >> 1;

}

private void buttonDCalculate\_Click(object sender, EventArgs e)

{

const int n = 7;

int s = 0;

int [] u = new int [n];

int [] d = new int [n];

int [] p = new int [n];

int[,] m = new int[n, n];

for (int sw = 0; sw < 3; sw++)

{

for (int i = 0; i < ribs.GetLength(0); i++)

{

u[i] = 0;

d[i] = inf;

p[i] = 0;

for (int j = 0; j < ribs.GetLength(1); j++)

m[i, j] = ribs[i, j];

}

d[sw] = 0;

u[s] = 0;

p[s] = s;

for (int i = 0; i < 4; i++)

{

int vertex = 0, min = inf;

for (int i1 = 0; i1 < n; i1++)

if (u[i1] == 0 && d[i1] < min)

{

min = d[i1];

vertex = i1;

}

if (vertex == n)

break;

for (int j = 3; j < n; j++)

if (u[j] == 0)

if (d[j] > d[vertex] + m[vertex, j])

{

d[j] = d[vertex] + m[vertex, j];

p[j] = vertex;

}

u[vertex] = 1;

for (int j = 0; j < 4; j++)

d2[sw, j] = d[j + 3];

for (int j = 0; j < 4; j++)

p2[sw, j] = p[j + 3] + 1;

//p2[sw, j] = p[j + 3] + 1;

}

/////////////

TNVDG.Rows[0].Cells[1].Value = ("B1").ToString();

TNVDG.Rows[0].Cells[2].Value = ("B2").ToString();

TNVDG.Rows[0].Cells[3].Value = ("B3").ToString();

TNVDG.Rows[0].Cells[4].Value = ("B4").ToString();

TNVDG.Rows[0].Cells[5].Value = ("ai").ToString();

TNVDG.Rows[1].Cells[0].Value = ("A1").ToString();

TNVDG.Rows[2].Cells[0].Value = ("A2").ToString();

TNVDG.Rows[3].Cells[0].Value = ("A3").ToString();

TNVDG.Rows[4].Cells[0].Value = ("bj").ToString();

for (int i = 1; i < 4; i++)

{

for (int j = 1; j < 5; j++)

TNVDG.Rows[i].Cells[j].Value = d2[i - 1, j - 1];

TNVDG.Rows[i].Cells[5].Value = TT[i - 1, 4];

}

for (int j = 0; j < 4;j++)

TNVDG.Rows[4].Cells[j + 1].Value = TT[3, j];

button2.Enabled = true;

}

buttonDCalculate.Enabled = false;

}

private void buttonDExit\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void pictureBoxDG\_MouseDoubleClick(object sender, MouseEventArgs e)

{

if (Application.OpenForms["Form2"] == null)

{

Form2 form2 = new Form2();

form2.Show();

}

}

private void button2\_Click(object sender, EventArgs e)

{

int oper = 0;

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("1").ToString();

calc.Rows[oper].Cells[4].Value = TT[0, 4].ToString();

calc.Rows[oper].Cells[9].Value = TT[1, 4].ToString();

calc.Rows[oper].Cells[14].Value = TT[2, 4].ToString();

calc.Rows[oper].Cells[15].Value = TT[3, 0].ToString();

calc.Rows[oper].Cells[16].Value = TT[3, 1].ToString();

calc.Rows[oper].Cells[17].Value = TT[3, 2].ToString();

calc.Rows[oper++].Cells[18].Value = TT[3, 3].ToString();

const int n = 3, m = 2;

int i = m;

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("2").ToString();

calc.Rows[oper++].Cells[20].Value = (i + 1).ToString();

int j = 0;

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("3").ToString();

calc.Rows[oper++].Cells[21].Value = (j + 1).ToString();

label1: int i\_ = i, j\_ = j;

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("4").ToString();

calc.Rows[oper].Cells[22].Value = (i\_ + 1).ToString();

calc.Rows[oper++].Cells[23].Value = (j\_ + 1).ToString();

if (TT[i, n + 1] <= TT[m + 1, j])

{

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("5").ToString();

calc.Rows[oper++].Cells[24].Value = ("Так").ToString();

TT[i, j] = TT[i, n + 1];

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("7").ToString();

calc.Rows[oper++].Cells[(n + 2) \* i + j].Value = TT[i, j].ToString();

}

else

{

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("5").ToString();

calc.Rows[oper++].Cells[24].Value = ("Ні").ToString();

TT[i, j] = TT[m + 1, j];

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("6").ToString();

calc.Rows[oper++].Cells[(n + 2) \* i + j].Value = (TT[i, j]).ToString();

}

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("8").ToString();

TT[i, n + 1] = TT[i, n + 1] - TT[i, j];

TT[m + 1, j] = TT[m + 1, j] - TT[i, j];

calc.Rows[oper].Cells[(n + 2) \* i + (n + 1)].Value = (TT[i, n + 1]).ToString();

calc.Rows[oper++].Cells[(n + 2) \* (m + 1) + j].Value = (TT[m + 1, j]).ToString();

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("9").ToString();

if (TT[i, n + 1] == 0)

{

calc.Rows[oper++].Cells[25].Value = ("Так").ToString();

i--;

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("10").ToString();

calc.Rows[oper++].Cells[20].Value = (i + 1).ToString();

}

else

{

calc.Rows[oper++].Cells[25].Value = ("Ні").ToString();

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("11").ToString();

if (TT[m + 1, j] == 0)

{

calc.Rows[oper++].Cells[26].Value = ("Так").ToString();

j++;

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("12").ToString();

calc.Rows[oper++].Cells[21].Value = (j + 1).ToString();

}

else

calc.Rows[oper++].Cells[26].Value = ("Ні").ToString();

}

if ((i\_ != 0) || (j\_ != n))

{

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("13").ToString();

calc.Rows[oper++].Cells[27].Value = ("Так").ToString();

goto label1;

}

else

{

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("13").ToString();

calc.Rows[oper++].Cells[27].Value = ("Ні").ToString();

calc.Rows.Add(1);

calc.Rows[oper].HeaderCell.Value = ("14").ToString();

calc.Rows[oper].Cells[0].Value = ("Σ=").ToString();

calc.Rows[oper].Cells[1].Value = (oper + 1).ToString();

for (int i1 = 0; i1 < TT.GetLength(0); i1++)

for (int j1 = 0; j1 < TT.GetLength(1); j1++)

TTG.Rows[i1].Cells[j1].Value = TT[i1, j1];

}

int C0 = 0;

for (int i1 = 0; i1 < 3; i1++)

for (int j1 = 0; j1 < 4; j1++)

C0 = C0 + TT[i1, j1] \* d2[i1, j1];

label7.Text = C0.ToString() + " у.г.о.";

button2.Enabled = false;

button4.Enabled = true;

button7.Enabled = true;

}

private void button1\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button3\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button4\_Click(object sender, EventArgs e)

{

int[,] X= TT;

for (int i = 0; i < 5; i++)

for (int j = 0; j < 6; j++)

if (i < 3 && j < 4)

if (TT[i, j] == 0)

X[i, j] = -1;

else X[i, j] = TT[i, j];

const int N = 3, M = 4;

bool stop,ok=false;

int [] a = new int[N];

int[] b = new int[M];

for (int j = 0; j < N; j++)

X[j, M] = a[j] = Convert.ToInt32(TNVDG.Rows[j + 1].Cells[5].Value);

for (int j = 0; j < M; j++)

X[N,j] = b[j] = Convert.ToInt32(TNVDG.Rows[4].Cells[j + 1].Value);

//////////////////////////////////

OPDG.Rows[0].Cells[1].Value = ("B1").ToString();

OPDG.Rows[0].Cells[2].Value = ("B2").ToString();

OPDG.Rows[0].Cells[3].Value = ("B3").ToString();

OPDG.Rows[0].Cells[4].Value = ("B4").ToString();

OPDG.Rows[0].Cells[5].Value = ("ai").ToString();

OPDG.Rows[1].Cells[0].Value = ("A1").ToString();

OPDG.Rows[2].Cells[0].Value = ("A2").ToString();

OPDG.Rows[3].Cells[0].Value = ("A3").ToString();

OPDG.Rows[4].Cells[0].Value = ("bj").ToString();

//////////////////////////////////

PODG.Rows[0].Cells[1].Value = ("B1").ToString();

PODG.Rows[0].Cells[2].Value = ("B2").ToString();

PODG.Rows[0].Cells[3].Value = ("B3").ToString();

PODG.Rows[0].Cells[4].Value = ("B4").ToString();

PODG.Rows[0].Cells[5].Value = ("Ui").ToString();

PODG.Rows[1].Cells[0].Value = ("A1").ToString();

PODG.Rows[2].Cells[0].Value = ("A2").ToString();

PODG.Rows[3].Cells[0].Value = ("A3").ToString();

PODG.Rows[4].Cells[0].Value = ("Vj").ToString();

//////////////////////////////////

bool [,] T = new bool [N,M];

for (int i = 0; i < N; i++)

for (int j = 0; j < M; j++)

T[i,j] = false;

int L = 0;

for (int i = 0; i < N; i++)

for (int j = 0; j < M; j++)

if (X[i,j] >= 0) L++;

int d = M + N - 1 - L;

int d1 = d;

/////////////

do

{

stop = true;

int [] u = new int [N];

int [] v = new int [M];

bool [] ub = new bool[N];

bool [] vb = new bool[M];

for (int i = 0; i < N; i++) ub[i] = false;

for (int i = 0; i < M; i++) vb[i] = false;

u[0] = 0;

ub[0] = true;

int count = 1;

int tmp = 0;

do

{

for (int i = 0; i < N; i++)

if (ub[i] == true)

for (int j = 0; j < M; j++)

if (X[i,j] >= 0)

if (vb[j] == false)

{

v[j] = d2[i,j] - u[i];

vb[j] = true;

count++;

}

for (int j = 0; j < M; j++)

if (vb[j] == true)

for (int i = 0; i < N; i++)

if (X[i,j] >= 0)

if (ub[i] == false)

{

u[i] = d2[i,j] - v[j];

ub[i] = true;

count++;

}

tmp++;

} while ((count < (M + N - d \* 2)) && (tmp < M \* N));

bool t = false;

if ((d > 0) || ok == false) t = true;

while (t)

{

for (int i = 0; (i < N); i++)

if (ub[i] == false)

for (int j = 0; (j < M); j++)

if (vb[j] == true)

{

if (d > 0)

if (T[i,j] == false)

{

X[i,j] = 0;

d--;

T[i,j] = true;

}

if (X[i,j] >= 0)

{

u[i] = d2[i,j] - v[j];

ub[i] = true;

}

}

for (int j = 0; (j < M); j++)

if (vb[j] == false)

for (int i = 0; (i < N); i++)

if (ub[i] == true)

{

if (d > 0)

if (T[i,j] == false)

{

X[i,j] = 0;

d--;

T[i,j] = true;

}

if (X[i,j] >= 0)

{

v[j] = d2[i,j] - u[i];

vb[j] = true;

}

}

t = false;

for (int i = 0; i < N; i++)

if (ub[i] == false) t = true;

for (int j = 0; j < M; j++)

if (vb[j] == false) t = true;

}

//-----------

int [,] D = new int [N,M];

for (int i = 0; i < N; i++)

for (int j = 0; j < M; j++)

{

if (X[i,j] >= 0)

D[i,j] = 88;

else

D[i,j] = d2[i,j] - u[i] - v[j];

if (D[i,j] < 0)

{

stop = false;

}

}

//

////////

if (stop == false)

{

int [,] Y = new int[N,M];

float find1, find2;

float best1 = 0;

float best2 = 0;

int ib1 = -1;

int jb1 = -1;

int ib2 = -1;

int jb2 = -1;

for (int i = 0; i < N; i++)

for (int j = 0; j < M; j++)

if (D[i,j] < 0)

{

for (int i1 = 0; i1 < N; i1++)

for (int j1 = 0; j1 < M; j1++)

Y[i1, j1] = 0;

find1 = find\_gor(i, j, i, j, N, M, X, Y, 0, -1);

for (int i1 = 0; i1 < N; i1++)

for (int j1 = 0; j1 < M; j1++)

Y[i1,j1] = 0;

find2 = find\_ver(i, j, i, j, N, M, X, Y, 0, -1);

if (find1 > 0)

if (best1 > D[i,j] \* find1)

{

best1 = D[i,j] \* find1;

ib1 = i;

jb1 = j;

}

if (find2 > 0)

if (best2 > D[i,j] \* find2)

{

best2 = D[i,j] \* find2;

ib2 = i;

jb2 = j;

}

}

if ((best1 == 0) && (best2 == 0))

{

ok = false;

d = d1;

for (int i = 0; i < N; i++)

for (int j = 0; j < M; j++)

if (X[i,j] == 0) X[i,j] = -1;

continue;

}

else

{

for (int i = 0; i < N; i++)

for (int j = 0; j < M; j++)

Y[i,j] = 0;

int ib, jb;

if (best1 < best2)

{

find\_gor(ib1, jb1, ib1, jb1, N, M, X, Y, 0, -1);

ib = ib1;

jb = jb1;

}

else

{

find\_ver(ib2, jb2, ib2, jb2, N, M, X, Y, 0, -1);

ib = ib2;

jb = jb2;

}

for (int i = 0; i < N; i++)

{

for (int j = 0; j < M; j++)

{

if ((X[i,j] == 0) && (Y[i,j] < 0))

{

stop = true;

ok = false;

d = d1;

}

X[i,j] = X[i,j] + Y[i,j];

if ((i == ib) && (j == jb)) X[i,j] = X[i,j] + 1;

if ((Y[i,j] <= 0) && (X[i,j] == 0)) X[i,j] = -1;

}

}

}

//

ok = true;

for (int i = 0; i < N; i++)

for (int j = 0; j < M; j++)

T[i,j] = false;

L = 0;

for (int i = 0; i < N; i++)

for (int j = 0; j < M; j++)

if (X[i,j] >= 0) L++;

d = M + N - 1 - L;

d1 = d;

if (d > 0) ok = false;

}

for (int i = 0; i < N; i++)

PODG.Rows[i + 1].Cells[5].Value = u[i].ToString();

for (int i = 0; i < M; i++)

PODG.Rows[4].Cells[i + 1].Value = v[i].ToString();

} while (stop == false);

///////////////////////////////////

for (int i = 1; i < N + 2; i++)

for (int j = 1; j < M + 2; j++)

if (X[i - 1, j - 1]< 0)

OPDG.Rows[i].Cells[j].Value = "∞";

else OPDG.Rows[i].Cells[j].Value = X[i - 1, j - 1];

for (int i = 1; i < N + 1; i++)

for (int j = 1; j < M + 1; j++)

if (X[i - 1, j - 1] < 0)

PODG.Rows[i].Cells[j].Value = "∞";

else PODG.Rows[i].Cells[j].Value = X[i - 1, j - 1];

//////////////////////

int C0 = 0;

for (int i = 0; i < N; i++)

for (int j = 0; j < M; j++)

if (X[i, j] > 0)

C0 = C0 + X[i, j] \* d2[i, j];

label8.Text = C0.ToString() + " у.г.о.";

button4.Enabled = false;

button5.Enabled = true;

}

public static int find\_gor(int i\_next,int j\_next,int im,int jm,int n,int m,int [,] X ,int [,] Y,int odd,int Xmin)

{

int rez=-1;

for(int j=0;j<m;j++)

if(((X[i\_next,j]>=0)&&(j!=j\_next))||((j==jm)&&(i\_next==im)&&(odd!=0)))

{

odd++;

if(odd>1000)

{

return -1;

}

int Xmin\_old=-1;

if((odd%2)==1)

{

Xmin\_old=Xmin;

if(Xmin<0)Xmin=X[i\_next,j];

else if((X[i\_next,j]<Xmin)&&(X[i\_next,j]>=0))

{

Xmin=X[i\_next,j];

}

}

if((j==jm)&&(i\_next==im)&&((odd%2)==0))

{

Y[im,jm]= Xmin;

return Xmin;

}

else rez=find\_ver(i\_next,j,im,jm,n,m,X,Y,odd,Xmin);

if(rez>=0)

{

if(odd%2==0)Y[i\_next,j]=Y[im,jm];

else Y[i\_next,j]=-Y[im,jm];

break;

}

else

{

odd--;

if(Xmin\_old>=0)

Xmin=Xmin\_old;

}

}

return rez;

}

////////////////////////

public static int find\_ver(int i\_next,int j\_next,int im,int jm,int n,int m,int [,] X,int [,] Y,int odd,int Xmin)

{

int rez=-1;

int i;

for(i=0;i<n;i++)

if(((X[i,j\_next]>=0))&&(i!=i\_next)||((j\_next==jm)&&(i==im)&&(odd!=0)))

{

odd++;

if(odd>1000)

{

return -1;

}

int Xmin\_old=-1;

if((odd%2)==1)

{

Xmin\_old=Xmin;

if(Xmin<0)Xmin=X[i,j\_next];

else if((X[i,j\_next]<Xmin)&&(X[i,j\_next]>=0))

Xmin=X[i,j\_next];

}

if((i==im)&&(j\_next==jm)&&((odd%2)==0))

{

Y[im,jm]= Xmin;

return Xmin;

}

else rez=find\_gor(i,j\_next,im,jm,n,m,X,Y,odd,Xmin);

if(rez>=0)

{

if(odd%2==0)Y[i,j\_next]=Y[im,jm];

else Y[i,j\_next]=-Y[im,jm];

break;

}

else

{

odd--;

if(Xmin\_old>=0)

Xmin=Xmin\_old;

}

}

return rez;

}

public static string num\_nam(int number)

{

string nstr="";

switch (number)

{

case 1: nstr = "A1"; break;

case 2: nstr = "A2"; break;

case 3: nstr = "A3"; break;

case 4: nstr = "B1"; break;

case 5: nstr = "B2"; break;

case 6: nstr = "B3"; break;

case 7: nstr = "B4"; break;

}

return nstr;

}

private void button5\_Click(object sender, EventArgs e)

{

richTextBoxDRs.AppendText("\t\tРезультат роботи програми\n");

richTextBoxDRs.AppendText("\t\tМетод Дейкстри\n");

richTextBoxDRs.AppendText("\t\tМатриця вхідних маршрутів має вигляд\n");

for (int i = 0; i < 7; i++)

{

richTextBoxDRs.AppendText("\t");

for (int j = 0; j < 7; j++)

richTextBoxDRs.AppendText("\t" + dataGridViewDM.Rows[i].Cells[j].Value);

richTextBoxDRs.AppendText("\n");

}

richTextBoxDRs.AppendText("\n");

//////////////////////////////////////

richTextBoxDRs.AppendText("\t\tМатриця найкоротших маршрутів має вигляд\n");

for (int i = 0; i < 5; i++)

{

richTextBoxDRs.AppendText("\t");

for (int j = 0; j < 6; j++)

richTextBoxDRs.AppendText("\t" + TNVDG.Rows[i].Cells[j].Value);

richTextBoxDRs.AppendText("\n");

}

richTextBoxDRs.AppendText("\n");

//////////////////////////////////////

richTextBoxDRs.AppendText("\t\tПобудова опорного плану методом Південно-Західного кута\n");

richTextBoxDRs.AppendText("\t\tОпорний план має вигляд:\n");

for (int i = 0; i < 3; i++)

{

richTextBoxDRs.AppendText("\t");

for (int j = 0; j < 4; j++)

richTextBoxDRs.AppendText("\t" + TTG.Rows[i].Cells[j].Value);

richTextBoxDRs.AppendText("\n");

}

richTextBoxDRs.AppendText("\t\t" + label6.Text + " " + label7.Text);

richTextBoxDRs.AppendText("\n");

richTextBoxDRs.AppendText("\t\tЗнаходження оптимального плану методом Потенціалів\n");

richTextBoxDRs.AppendText("\t\t\t" + label14.Text + "\n");

for (int i = 0; i < 5; i++)

{

richTextBoxDRs.AppendText("\t");

for (int j = 0; j < 6; j++)

richTextBoxDRs.AppendText("\t" + OPDG.Rows[i].Cells[j].Value);

richTextBoxDRs.AppendText("\n");

}

richTextBoxDRs.AppendText("\t\t\t" + label13.Text + "\n");

for (int i = 0; i < 5; i++)

{

richTextBoxDRs.AppendText("\t");

for (int j = 0; j < 6; j++)

richTextBoxDRs.AppendText("\t" + PODG.Rows[i].Cells[j].Value);

richTextBoxDRs.AppendText("\n");

}

richTextBoxDRs.AppendText("\t\t" + label9.Text + " " + label8.Text);

button5.Enabled = false;

button6.Enabled = true;

}

private void button6\_Click(object sender, EventArgs e)

{

saveFileDialog.Filter = "txt файли(\*.txt)|\*.txt";

saveFileDialog.FileName = "Результати";

if (saveFileDialog.ShowDialog() == DialogResult.OK)

{

richTextBoxDRs.SaveFile(saveFileDialog.FileName, RichTextBoxStreamType.PlainText);

}

}

private void button7\_Click(object sender, EventArgs e)

{

Word.Application application = new Word.Application();

Object missing = Type.Missing;

application.Documents.Add(ref missing, ref missing, ref missing, ref missing);

Word.Document document = application.ActiveDocument;

Word.Range range = application.Selection.Range;

Object behiavor = Word.WdDefaultTableBehavior.wdWord9TableBehavior;

Object autoFitBehiavor = Word.WdAutoFitBehavior.wdAutoFitFixed;

document.Tables.Add(document.Paragraphs[1].Range, calc.Rows.Count, calc.Columns.Count, ref behiavor, ref autoFitBehiavor);

for (int i = 0; i < calc.Rows.Count; i++)

for (int j = 0; j < calc.Columns.Count; j++)

if (calc.Rows[i].Cells[j].Value != null)

document.Tables[1].Cell(i + 1, j + 1).Range.Text = calc.Rows[i].Cells[j].Value.ToString();

else document.Tables[1].Cell(i + 1, j + 1).Range.Text = " ".ToString();

application.Visible = true;

}

}

}

# **Висновок**

Теорія графів посідає досить важливе місце не тільки у математиці, але й у багатьох інших дисциплінах та в різних сферах життєдіяльності. Простота графових моделей, їх наочність у представленні дозволяє розв'язувати складні задачі і слідкувати за їх розв'язком не тільки на всьому графі, а й на окремих його частинах.

Задачі оптимізації - широкий і важливий розділ в математиці, але іноді виникають труднощі в застосуванні стандартних підходів до їх вирішення. Тоді до їх розв'язання можна застосувати простий, але ефективний апарат теорії графів. Для розв’язання різних типів задач можна скористатися різними методами, наприклад при розв’язанні задач про пошук найкоротших шляхів між заданими множинами вершин можна скористатися модифікованим методом Дейкстри, оскільки звичайний метод Дейкстри недостатній для її рішення, а метод Флойда – надлишковий.

Також для побудови опорного плану перевезень можна скористатися одним із 12 методів, кожен з яких має свої переваги та недоліки.

# **Список використаної літератури**

1. ГлинськийЯ.М. TurboPascalіDelphi / Я.М. Глинський, В.Є. Анохін, В.А. Ряжська // Навч. посіб. 9-те доп. вид. – Львів: Деол, СПД Глинський, 2009. – 192 с.
2. Бабушкина И.Практикум по объектно-ориентированному программированию / И.Бабушкина, С.Окулов. – М.: Бином: Лаборатория знаний, 2004. – 299 с.
3. Мануйлов В.Г.Разработка программного обеспечения на Паскале / В.Г.Мануйлов. – М.: 2005. – 240 с.
4. Прокудін Г.С. Модифікація методу Дейкстри стосовно розв’язання сітьових транспортних задач / Г.С. Прокудін // Вісник НТУ та ТАУ, № 7. – Київ : НТУ, 2002. – С. 195–198
5. Четверухін Б.М. Оптимізація перевезень вантажів у сітьової постановці / Б.М. Четверухін, Г.С. Прокудін, С.О. Білоус // Вісник НТУ та ТАУ, № 8. – Київ : НТУ, 2003. – С. 93–100
6. Прокудін Г.С. Модель комплексных перевозок в транспортных системах / Г.С. Прокудін // Проблемиінформатизації та управління, вип. 12. – К.: НАУ, 2005. – С. 138–142
7. Прокудін Г.С. Приклад організації вантажних перевезень на транспортної мережі України / Г.С. Прокудін // Управління проектами, системнийаналіз і логістика, вип. 5. – К.: НТУ, 2008. – С. 150–158
8. Прокудін Г.С. Моделі та методи оптимізації вантажних перевезень в транспортних системах / / Г.С. Прокудін // Автореф. докт. дисер. – К.: НТУ, 2009. – 43 с.
9. Прокудін Г.С. Інформаційна система оптимізації вантажних перевезень в транспортних системах / Г.С. Прокудін, В.Д. Данчук, О.Г. Прокудін // Проблеми транспорту, вип. 6. – К.: НТУ, 2010. – С. 90–95