

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**Інститут КНІТ
Кафедра ПЗ**

ЗВІТ

До лабораторної роботи №3

На тему: *“Розв’язування задачі цілочислового програмування за допомогою методу Гоморі. Геометрична інтерпретація розв’язку”*

З дисципліни: *“Дослідження операцій”*

Лектор:

Доктор технічних наук
Журавчак Л. М.

Виконав:

студент групи ПЗ-22
Павлів М. Я.

Прийняв:

Кандидат ф.-м. наук
Івасько Н. М.

«__»_____ 2022 р.

Σ = _____.

Львів – 2022

Тема роботи: розв'язування задачі цілочислового програмування за допомогою методу Гоморі. Геометрична інтерпретація розв'язку.

Мета роботи: ознайомитись на практиці із основними алгоритмами розв'язування цілочислових задач математичного програмування.

Лабораторне завдання

1. Отримати індивідуальний варіант завдання.
2. Написати програму розв'язування задачі ЦЛП методом Гоморі та знайти розв'язок (максимальне значення функції та значення цілочислових змінних, при якому воно досягається) згідно з варіантом Додатку 1 до лабораторної роботи № 3.
3. Розв'язати задачу з Додатка 1 до лабораторної роботи № 3 графічним методом (знайти максимальне значення функції та значення змінних, при якому воно досягається).
4. Оформити звіт про виконану роботу.
5. Продемонструвати викладачеві результати, відповісти на запитання стосовно виконання роботи.

Індивідуальне завдання

14	$f(x_1, x_2) = 8x_1 + 5x_2 \rightarrow \max$ <p>Обмеження:</p> $\begin{cases} 5x_1 + 2x_2 \leq 20; \\ x_1 + x_2 \leq 6; \\ x_1, x_2 \geq 0; x_1, x_2 \in Z. \end{cases}$
----	--

Теоретичні відомості

14. Запишіть нерівність, яка визначає правильне відсікання.

Припустимо, що є наступне рівняння, для якого треба ввести додаткове обмеження:

$$x_1 + \frac{3}{10}x_4 - \frac{1}{4}x_5 = \frac{7}{2}.$$

Додаткове обмеження (за методом Гоморі для частково цілочислових задач) має вигляд:

$$\frac{3}{10}x_4 + \frac{1}{4}x_5 \geq \frac{1}{2}$$

3. Які є типи задач ЦЛП?

- Повністю цілочислові задачі: всі змінні повинні мати цілочисловий розв'язок.
- Частково цілочислові задачі: такі задачі вимагають, щоб принаймні одна змінна повинна мати цілочисловий розв'язок.

7. Назвіть основні кроки другого алгоритму Гоморі.

Крок 1. Використовуючи будь-який з варіантів СМ, знаходимо оптимальний розв'язок задачі без урахування умов цілочисловості. Якщо задача не має розв'язку – кінець. Якщо для отриманого розв'язку виконуються умови цілочисловості, то кінець – знайдений оптимальний розв'язок задачі.

Крок 2. Укладаємо додаткове обмеження для змінної, яка в поточному оптимальному розв'язку має максимальну дробову частину та повинна бути цілочисловою згідно з умовами задачі за правилами:

$$\sum_{j=1}^k \gamma_{lj} x_j \geq \beta_l,$$

де величини γ_{lj} визначаємо з таких співвідношень:

1) для нецілочислових значень змінних x_j :

$$\gamma_{lj} = \begin{cases} a_{lj}, & \text{якщо } a_{lj} \geq 0, \\ \frac{\beta_l}{1 - \beta_l} |a_{lj}|, & \text{якщо } a_{lj} < 0; \end{cases}$$

2) для цілочислових змінних x_j :

$$\gamma_{lj} = \begin{cases} \alpha_{lj}, & \text{якщо } \alpha_{lj} \leq \beta_l, \\ \frac{\beta_l}{1 - \beta_l} (1 - \alpha_{lj}), & \text{якщо } \alpha_{lj} > \beta_l. \end{cases}$$

Крок 3. За допомогою двоїстого СМ знаходимо оптимальний розв'язок задачі з додатковим обмеженням.

Крок 4. Якщо для отриманого розв'язку виконуються умови цілочисловості, то кінець – знайдений оптимальний розв'язок задачі. Якщо розв'язок задачі відсутній – кінець. В іншому випадку переходимо до кроку 2.

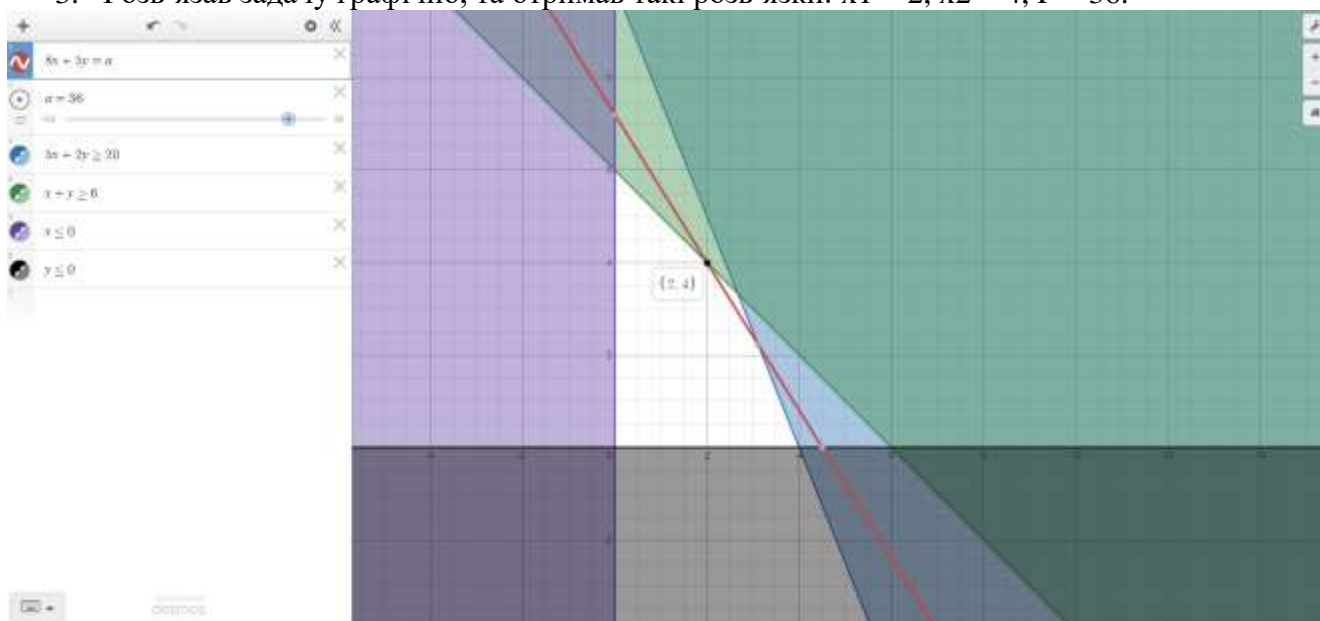
Хід роботи

1. Написав програму для частково цілочислових задач на мові C#. Для методу Гоморі використовував раніше розроблені симплекс метод та двоїстий симплекс метод.
2. Розв'язав задачу за допомогою програми, отримав такі розв'язки:

xb	cb	P0	c1=8.000	c2=5.000	c3=0.000	c4=0.000	c5=0.000	c6=0.000	c7=0.000
x1	8.000	2.000	1.000	0.000	0.000	0.000	-4.000	0.000	1.000
x2	5.000	4.000	0.000	1.000	0.000	0.000	6.500	0.000	-1.500
x4	0.000	0.000	0.000	0.000	0.000	1.000	-2.500	0.000	0.500
x3	0.000	2.000	0.000	0.000	1.000	0.000	7.000	0.000	-2.000
x6	0.000	1.000	0.000	0.000	0.000	0.000	4.000	1.000	-1.500
Q	=	36.000	0.000	0.000	0.000	0.000	0.500	0.000	0.500

X1=2, X2=4; F = 36.

3. Розв'язав задачу графічно, та отримав такі розв'язки: x1 = 2, x2 = 4, F = 36.



Код програми

Program.cs

```
var processor = new SimplexProcessor(new ConsoleReader());
Tuple<SimplexFunction, SimplexConstraint[]> data = processor.GetData();
```

```

var function = data.Item1;
var constraints = data.Item2;

var gomoryMethod = new GomoryMethod(function, constraints);
List<DualSimplexSnapshot> steps = gomoryMethod.Solve().ToList();
ISimplexTableBuilder builder = new ConsoleSimplexTableBuilder(steps);
List<string> tables = builder.Build('x', 'Q').ToList();

foreach (var table in tables)
{
    Console.WriteLine(table);
}

```

GomoryMethod.cs

```

using Lab1;
using Lab2;

namespace Lab3
{
    public class GomoryMethod
    {
        private const int MaxAllowedSteps = 50;
        private List<double> _initialFCoefficients;
        private List<double> _functionCoefficients;
        private double[,] _systemMatrix;
        private int _mainVariablesCount;
        private List<double> _b;
        private double _q;
        private int _rowCount;
        private int _columnCount;
        private List<int> _basisIndices;

        public GomoryMethod(
            SimplexFunction function,
            SimplexConstraint[] constraints
        )
        {
            if (constraints.GroupBy(constraint => constraint.Count).First().Count() !=
constraints.Length)
            {
                throw new Exception();
            }

            if (function.VarCount != constraints.First().Count)
            {
                throw new Exception();
            }

            function = function.ToCanonical();
            constraints = constraints.ToCanonical().ToArray();

            _mainVariablesCount = function.VarCount;

            _initialFCoefficients = Enumerable
                .Range(0, constraints.First().Count)
                .Select(i =>
                {
                    if (i < function.VarCount)
                    {
                        return function[i];
                    }

                    return 0;
                }).ToList();

            _functionCoefficients = _initialFCoefficients

```

```

        .Select(coefficient => -coefficient)
        .ToList();

    _systemMatrix = constraints.ToMatrix();

    _b = constraints
        .Select(c => c.Value)
        .ToList();

    _q = 0;
    _rowCount = _systemMatrix.GetLength(0);
    _columnCount = _systemMatrix.GetLength(1);
    _basisIndices = Enumerable.Range(_mainVariablesCount, _columnCount -
    _mainVariablesCount).ToList();
    }

    public IEnumerable<DualSimplexSnapshot> Solve()
    {
        List<DualSimplexSnapshot> steps = new Simplex(
            _systemMatrix,
            _functionCoefficients.ToArray(),
            _b.ToArray())
            .Solve().ToList();

        if (IsSolutionInteger(steps.Last()))
        {
            return steps;
        }

        SolveWithDualSimplex(steps);

        return steps;
    }

    private void SolveWithDualSimplex(List<DualSimplexSnapshot> steps)
    {
        do
        {
            var lastStep = steps.Last();
            SetSettings(lastStep);

            var rowIndex = GetLargestDecimalIndex();
            var newCoefficients = new double[_columnCount];
            var beta = _b[rowIndex] % 1;

            SetNewRow(newCoefficients, rowIndex, beta);

            var temp = 0;
            var newConstraint = new SimplexConstraint(
                newCoefficients, ">=", beta
            ).ToCanonical(1, ref temp);

            AppendNewRow(newConstraint);

            steps.AddRange(new DualSimplex(
                _systemMatrix,
                _functionCoefficients.ToArray(),
                _initialFCoefficients.ToArray(),
                _b.ToArray(),
                _q,
                _basisIndices
            ).Solve());

        } while (IsSolutionInteger(steps.Last()) == false && steps.Count <
MaxAllowedSteps);
    }

```

```

}

private void SetNewRow(double[] newCoefficients, int rowIndex, double beta)
{
    for (var i = 0; i < newCoefficients.Length; ++i)
    {
        var element = _systemMatrix[rowIndex, i];

        if (IsIndexOfMainVariable(i) == false)
        {
            newCoefficients[i] = element.IsGreaterThanOrEqualWithin(0, 1e-8)
                ? element
                : (Math.Abs(element) * beta) / (1 - beta);
            continue;
        }

        var alpha = element % 1;

        newCoefficients[i] = alpha.IsLessThanOrEqualWithin(beta, 1e-8)
            ? alpha
            : ((1 - alpha) * beta) / (1 - beta);
    }
}

private void AppendNewRow(SimplexConstraint newConstraint)
{
    _systemMatrix = _systemMatrix.Resize(_rowCount + 1, _columnCount + 1);
    _rowCount++;
    _columnCount++;

    for (var i = 0; i < _columnCount; ++i)
    {
        _systemMatrix[_rowCount - 1, i] = newConstraint[i];
    }

    _b.Add(newConstraint.Value);
    _basisIndices.Add(newConstraint.Count - 1);
    _initialFCoefficients.Add(0);
    _functionCoefficients.Add(0);
}

private void SetSettings(DualSimplexSnapshot step)
{
    _systemMatrix = step.SystemMatrix;
    _b.Clear();
    _b.AddRange(step.B);
    _functionCoefficients.Clear();
    _functionCoefficients.AddRange(step.FunctionCoefficients);
    _basisIndices.Clear();
    _basisIndices.AddRange(step.BasisIndices);
    _q = step.G;
}

private int GetLargestDecimalIndex()
{
    return _b
        .Select((b, i) => new { Value = b, Index = i })
        .Where(x => _basisIndices[x.Index] < _mainVariablesCount)
        .OrderBy(x => Math.Abs(x.Value % 1))
        .Last().Index;
}

private bool IsSolutionInteger(DualSimplexSnapshot step)
{
    return step.B

```

```

        .Where((_, i) => IsIndexOfMainVariable(_basisIndices[i]))
        .Count(b => (b % 1).IsLessThanOrEqualToWithin(1e-6, 1e-8)) ==
_mainVariablesCount;
    }

    private bool IsIndexOfMainVariable(int i)
    {
        return i < _mainVariablesCount;
    }
}

```

Висновки

На лабораторній роботі я на практиці ознайомився із основними алгоритмами розв'язування цілочислових задач математичного програмування. Розв'язав свій варіант за допомогою методу Гоморі для частково цілочислової задачі та звірів результати, розв'язавши задачу графічно. Розв'язки збіглись.