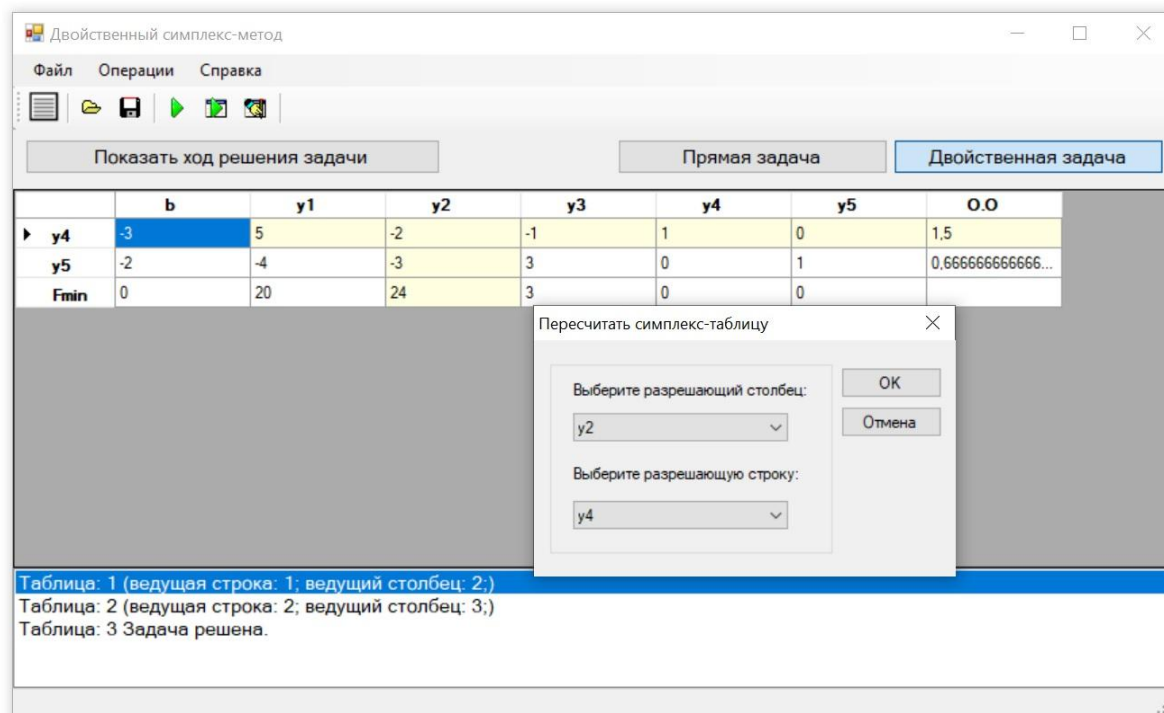


Двойственный симплекс-метод

Программа для решения задач линейного программирования двойственным симплекс-методом.



(с) Соколов А.П.

Связь с автором: alex_ey@mail.ru

Домашняя страница программы: <http://www.mathzone.ru/programs/11/>

Тип распространения: FreeWare

Если Вы нашли какие-нибудь ошибки в программе, или у Вас есть предложения и замечания по поводу данного продукта, то просьба сообщать о них, на мой электронный адрес.

Описание программы

Программа сама составляет двойственную задачу и приводит ее к каноническому виду. Выводит подробный отчет о ходе решения задачи. Есть возможность решения как двойственной, так и прямой задачи, и быстрое переключения между ними, для более подробного анализа и исследования задачи.

Имеется три режима решения задач:

- 1) Автоматический
- 2) Пошаговый
- 3) Ручной

В первом режиме программа сама выбирает разрешающий столбец и строку, которые обеспечивают максимальное возрастание или уменьшение целевой функции. А также автоматически пересчитывает все таблицы.

В пошаговом режиме, каждая пересчитанная таблица выводится на экран. Этот режим удобен для просмотра промежуточных результатов решения задачи. Разрешающий столбец и строку, в данном режиме, программа также выбирает сама.

В ручном режиме пользователь сам выбирает разрешающую строку и столбец.

Пример задачи на максимизацию

Завод выпускает продукцию 1-го и 2-го типа. Прибыль от реализации единицы продукции соответственно составляет 30 и 40 у.е. На выпуск единицы продукции 1-го типа расходуется 4 единиц сырья категории А, 4 ед. – категории В. Для выпуска единицы продукции 2-го типа расходуется сырья категории А - 3 ед., категории С – 12 единицы. Имеющиеся в наличие запасы сырья категории А – 120 единиц, В – 252 единицы.

Тип выпускаемой продукции	Расход сырья (ед.)		Прибыль от реализации единицы продукции (у.е.)
	А	В	
1	4	4	30
2	3	12	40
Запасы сырья (ед.)	120	252	

Необходимо определить количество продукции, при выпуске которой прибыль является максимальной.

Предположим, что будет изготовлено x_1 единиц продукции 1-го типа, x_2 – 2-го типа. Тогда для производства такого количества изделий потребуется затратить:

$4x_1 + 4x_2$ сырья вида А

Так как запас сырья данного вида не может превышать 120, то должно выполняться неравенство:

$$4x_1 + 4x_2 \leq 120$$

Аналогичные рассуждения относительно возможного использования сырья вида В приведут к следующим неравенствам:

$$3x_1 + 12x_2 \leq 252$$

При этом так как количество выпускаемой продукции не может быть отрицательной, то:

$$x_1 > 0, x_2 > 0. \quad (1)$$

Далее, если будет выпущено x_1 единиц продукции 1-го типа, x_2 единиц продукции 2-го типа, то прибыль от их реализации составит:

$$F = 30x_1 + 40x_2$$

Таким образом, приходим к следующей математической задаче:

$$4x_1 + 4x_2 \leq 120$$

$$3x_1 + 12x_2 \leq 252 \quad (2)$$

двух линейных неравенств с двумя неизвестными x_j ($j=1..2$) и линейная функция относительно этих же переменных:

$$F = 30x_1 + 40x_2 \quad (3)$$

требуется среди всех неотрицательных решений системы неравенств (2) найти такое, при котором функция (3) принимает максимальное значение.

Линейная функция (3), максимум которой требуется определить, вместе с системой неравенств (2) и условием неотрицательности переменных (1) образуют математическую модель исходной задачи. Так как функция (3) линейная, а система (2) содержит только линейные неравенства, то задача (1)-(3) является задачей линейного программирования.

Вводим получившуюся задачу в диалоговое окно и жмем "ОК".

Новая задача

Кэф-ты ограничений:			Свободные члены:	
4	4	<=	120	
3	12	<=	252	

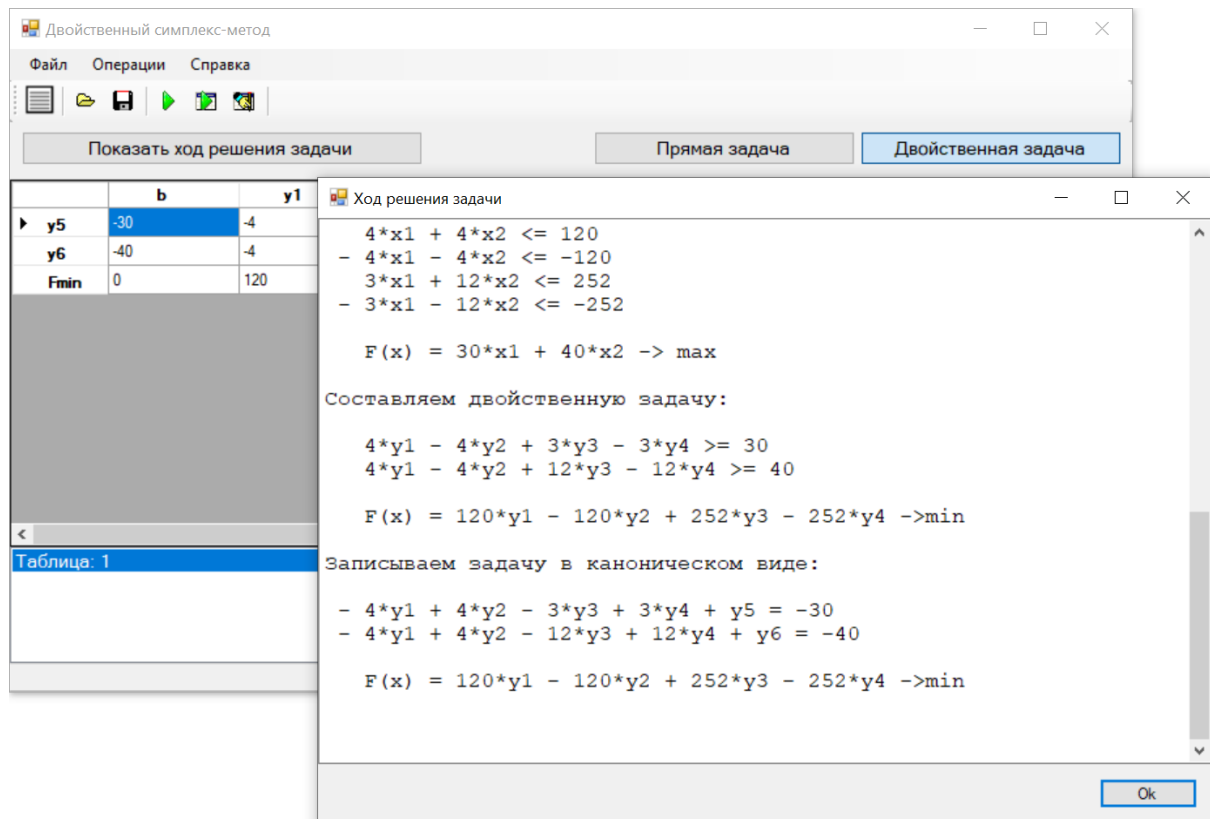
Кэф-ты целевой функции:

30	40
----	----

Максимизировать

OK Отмена

Далее программа сама приводит и составляет двойственную задачу, и приводит ее к каноническому виду:



В соответствии с приведенной к каноническому виду задачей формируется симплекс таблица:

	b	y1	y2	y3	y4	y5	y6	0.0
y5	-30	-4	4	-3	3	1	0	
y6	-40	-4	4	-12	12	0	1	
Fmin	0	120	-120	252	-252	0	0	

В ходе решения были получена следующая таблица:

	b	y1	y2	y3	y4	0.0
y1	6,666666666	1	0	-0,333333333	0,083333333	
y2	1,111111111	0	1	0,111111111	-0,111111111	
Fmin	-1080	0	0	12	18	

Базисным переменным y_1 , y_2 – присваиваем значения свободных членов. Остальным переменным присваиваем нули.

$$y_1 = 6,66666$$

$$y_2 = 1,11111$$

Значение целевой функции показывается, в левом нижнем углу таблицы. Так как исходной задачей был поиск минимума, оптимальное решение есть свободный член строки F, взятый с противоположным знаком.

$$F_{\min} = 1080$$

Курсовая работа + Исходники

Курсовая работа в ходе которой, была спроектирована, реализована и протестирована программа для решения задач линейного программирования симплекс-методом.

Программа написана на Microsoft Visual C#, исходники снабжены подробными комментариями. Курсовой был сдан на 5 (отлично).

Лист содержания курсового проекта:

СОДЕРЖАНИЕ	
ВВЕДЕНИЕ	3
1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ	7
1.1. Описание предметной области	7
1.2. Анализ предметной области.....	10
2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА.....	13
2.1. Выбор модели разработки ПО	13
2.2. Концептуальная модель программы	13
2.3. Разбиение программы на модули	14
2.4. Организация данных в программе.....	14
2.5. Логическая модель программы.....	16
3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА	18
3.1. Выбор языка и среды разработки	18
3.2. Описание алгоритма программы	23
3.3. Интерфейс приложения	27
4. ТЕСТИРОВАНИЕ И ОТЛАДКА ПРОЕКТА.....	32
5. ЗАКЛЮЧЕНИЕ	33
6. СПИСОК ЛИТЕРАТУРЫ.....	34
7. ПРИЛОЖЕНИЕ. Листинг программы.....	35

ХГУ ИИиТ. 13 Соколов Алексей Петрович	лист
	2

Фрагмент исходного кода программы:

```
// Процедура пересчета симплекс-таблицы,
// с учетом выбранного разрешающего столбца и
// разрешающей строки.
public void RecalcTable(int apivotRow, int apivotCol)
{
    // Каждый элемент разрешающей строки (строки в которой находится разрешающий элемент),
    // делим на разрешающий элемент
    double relement = table[apivotRow - 1, apivotCol];
    for (int i = 0; i < cols; i++)
    {
        table[apivotRow - 1, i] = table[apivotRow - 1, i] / relement;
    }

    // Все остальные элементы рассчитываются по формуле:
    // Для этого мы должны сделать равными 0
    // все остальные элементы разрешающего столбца таблицы 1,
    // кроме стоящего на пересечении с разрешающей строкой.
    // Домножаем на необходимое число разрешающую строку таблицы 1 и
    // прибавляем к другим строкам. Получаем симплекс-таблицу 2.
    for (int i = 0; i < rows; i++)
    {
        if (i != apivotRow - 1)
        {
            double temp = table[i, apivotCol];
            for (int j = 0; j < cols; j++)
            {
                double value = table[apivotRow - 1, j] * temp;
                table[i, j] = table[i, j] - value;
            }
        }
    }

    baseVector[apivotRow - 1] = apivotCol;

    GetRoots();
}

// Функция находит вектор корней
public void GetRoots()
{
    for (int i = 0; i < n; i++)
    {
        FRootVector[i] = 0;
    }

    for (int i = 0; i < m; i++)
    {
        if ((int)baseVector[i] <= n)
        {
            FRootVector[(int)baseVector[i] - 1] = table[i, 0];
        }
    }
}
```


Вы можете купить исходники программы за 299 рублей. Или исходники + курсовая и блок-схемы за 499 рублей.

Программа, Исходники **299 Рублей.**

<https://plati.market/itm/dual-simplex-method-c-sources/3402934>

Программа, Исходники, Курсовая, Блок-схемы **499 Рублей.**

<https://plati.market/itm/dual-simplex-method-coursework-c-sources/3402928>

Вы можете купить эту работу онлайн, т.е. прямо сейчас. Если Вы хотите приобрести эту работу другим способом, или у Вас есть какие-либо вопросы, касающиеся этой программы, то можете связаться со мной по email: alex_ey@mail.ru