ДНЕПРОПЕТРОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ ОЛЕСЯ ГОНЧАРА

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ
КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Лабораторная робота

на тему: «Решение задач линейного программирования. На примере задач минимизации потребления ресурсов на предприятии»

Выполнил: студент V курса, группы ПК-16м-1

6.040302 «Информатика»

Егошкин Данила Игоревич

г. Днепропетровск - 2016 року

Рассмотрим пример задачи из книги: «Справочник по экономико-математическим моделям и методам» автор Крушевский А. В.

По ходу задачи, я буду её дополнять объяснениями ☺

Страница 18, пример 7:

На предприятие поступают однотипные рулоны шириной 700 см, которые надо разрезать на заготовки трех видов:
1-й – шириной 230 см;
2-й – шириной 190 см;
3-й – шириной 80 см;

План заготовок следующий:
$a_1 = 60$ штук,
$a_2 = 90$ штук,
$a_3 = 320$ штук

План должен выполняться с минимальными отходами. С этой целью составляют варианты j-го раскроя рулонов на заготовки и определяют a_{ij} количество заготовок i-го вида из одного рулона, получаемых согласно j-му варианту, и отходы c_j. Эти данные приведены в таблице 1.

1. Показатели раскроя рулонов

Вариант раскроя	Число заготовок из одного рулона			Отходы см.
	1-го вида	2-го вида	3-го вида	
1	3	-	-	10
2	2	1	-	50
3	2	-	3	0
4	1	2	1	10
5	1	-	5	70
6	1	1	3	40
7	2	-	3	0
8	-	-	8	60
9	-	2	4	0

Математическая модель:

$$10x_1 + 50x_2 + 0 * x_3 + 10x_4 + 70x_5 + 40x_6 + 0 * x_7 + 60x_8 + 0 * x_9 \to min$$

Если вы не заметили, то **математическая модель** построена из столбца отходов. И это верно, ведь именно отходы мы ходим минимизировать. Так как, чем меньше отходов будет, тем меньше ресурсов нам понадобиться на производство единицы товара.

Продолжим описывать модель.

Математическая модель:

$$10x_1 + 50x_2 + 0 * x_3 + 10x_4 + 70x_5 + 40x_6 + 0 * x_7 + 60x_8 + 0 * x_9 \to min$$

При ограничения на план заготовок:

$$\begin{cases} 3x_1 + 2x_2 + 2x_3 + 1x_4 + 1x_5 + 1x_6 + 2x_7 + 0 * x_8 + 0 * x_9 = 60 \\ 0 * x_1 + 1x_2 + 0 * x_3 + 2x_4 + 0 * x_5 + 1x_6 + 0 * x_7 + 0 * x_8 + 2x_9 = 90 \\ 0 * x_1 + 0 * x_2 + 3x_3 + 1x_4 + 5x_5 + 3x_6 + 3x_7 + 8x_8 + 4x_9 = 320 \end{cases}$$

При этом: $x_i \geq 0, i = 1..9$

Если вы не заметили, то **ограничения на план заготовок** построены из столбца «число заготовок из одного рулона» и самого «плана заготовок». То есть:

$$\begin{cases} \text{число заготовок } 1-\text{го вида} = a_1 = 60 \text{ штук} \\ \text{число заготовок } 2-\text{го вида} = a_2 = 90 \text{ штук} \\ \text{число заготовок } 3-\text{го вида} = a_3 = 320 \text{ штук} \end{cases}$$

Далее в книге написано решение этой задачи, а именно сам ответ:

Решением этой задачи является:

$$x_1 = x_2 = x_4 = x_5 = x_6 = x_7 = 0$$

$$x_3 = 30$$

$$x_8 = 6{,}25$$

$$x_9 = 45$$

Отсюда следует, что минимальное количество отходов получается, если раскроить:

30 рулонов – по 3-му варианту (2 заготовки 1-го вида и 3 заготовки 3-го вида смотреть таблицу 2, выделено желтым);

6,25 рулона – раскроить по 8-му варианту (8 заготовок 3-го вида смотреть таблицу 2, выделено зеленым (светло-зеленым));

45 рулонов – по 9-му варианту (2 заготовки 2-го вида и 4 заготовки 3-го вида смотреть таблицу 2, выделено синим (светло-синим));

2. Лучшие варианты раскроя

Вариант раскроя	Число заготовок из одного рулона			Отходы см.
	1-го вида	2-го вида	3-го вида	
1	3	-	-	10
2	2	1	-	50
3	2	-	3	0
4	1	2	1	10
5	1	-	5	70
6	1	1	3	40
7	2	-	3	0
8	-	-	8	60
9	-	2	4	0

Тогда необходимо 30+45+6,25=81,25 рулона, а отходы составляют 6,25*60=375 см.

Теперь начнем разбирать, каким образом произошло такое решение задачи. Для этого вспомним курс «Методы оптимизаций» - если у вас не было такого курса, то дальше я буду рассказывать как можно решить эту задачу. Эта задача относиться к классу задач линейного программирования.

Линейное программирование

Формальные определения

Линейное программирование — математическая дисциплина, посвящённая теории и методам решения экстремальных задач на множествах n-мерного векторного пространства, задаваемых системами линейных уравнений и неравенств.

Линейное программирование (ЛП) является частным случаем выпуклого программирования, которое в свою очередь является частным случаем математического программирования. Одновременно оно — основа нескольких методов решения задач целочисленного и нелинейного программирования. Одним из обобщений линейного программирования является дробно-линейное программирование.

Многие свойства задач линейного программирования можно интерпретировать также как свойства многогранников и таким образом геометрически формулировать и доказывать их.

Типы задач

Общей (стандартной) задачей линейного программирования называется задача нахождения минимума линейной целевой функции (линейной формы) вида:

$$f(x) = \sum_{j=1}^{n} c_j x_j = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n$$

Задача, в которой фигурируют ограничения в форме неравенств, называется основной задачей линейного программирования (ОЗЛП):

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i \quad (i = 1,2, \dots, m)$$

$$x_j \geq 0 \quad (j = 1,2, \dots, n)$$

Задача линейного программирования будет иметь канонический вид, если в основной задаче вместо первой системы неравенств имеет место система уравнений с ограничениями в форме равенства:

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \ \ (i = 1,2,\dots,m)$$

Основную задачу можно свести к канонической путём введения дополнительных переменных.

Выводы

Мы имеем дело со стандартной задачей линейного программирования канонического вида.

Осталось понять как был получен ответ ☺

Для этого будем использовать наиболее известным и широко применяемым на практике для решения общей задачи линейного программирования (ЛП) - симплекс-метод.

Несмотря на то, что симплекс-метод является достаточно эффективным алгоритмом, показавшим хорошие результаты при решении прикладных задач ЛП, он является алгоритмом с экспоненциальной сложностью. Причина этого состоит в комбинаторном характере симплекс-метода, последовательно перебирающего вершины многогранника допустимых решений при поиске оптимального решения.

Существуют и другие методы решения подобных задач(М-задача, Метод Гомори, Двойственный симплекс-метод), но этот самый простой(хотя это относительно) и чаще всего используется, а так же объясняется во многих книгах ☺

Симплекс-метод

Симплекс-метод — алгоритм решения оптимизационной задачи линейного программирования путём перебора вершин выпуклого многогранника в многомерном пространстве.

Сущность метода: построение базисных решений, на которых монотонно убывает линейный функционал, до ситуации, когда выполняются необходимые условия локальной оптимальности.

Для нашей задачи необходим симплекс-метод ориентированный на минимизацию функционала(чаще всего используются нахождение максимума, а не минимума). Так же, наша задача уже приведена к каноническому виду, то есть у нас в ограничениях равенства:

Математическая модель:

$$10x_1 + 50x_2 + 0 * x_3 + 10x_4 + 70x_5 + 40x_6 + 0 * x_7 + 60x_8 + 0 * x_9 \to min$$

При ограничения на план заготовок:

$$\begin{cases} 3x_1 + 2x_2 + 2x_3 + 1x_4 + 1x_5 + 1x_6 + 2x_7 + 0 * x_8 + 0 * x_9 = 60 \\ 0 * x_1 + 1x_2 + 0 * x_3 + 2x_4 + 0 * x_5 + 1x_6 + 0 * x_7 + 0 * x_8 + 2x_9 = 90 \\ 0 * x_1 + 0 * x_2 + 3x_3 + 1x_4 + 5x_5 + 3x_6 + 3x_7 + 8x_8 + 4x_9 = 320 \end{cases}$$

При этом: $x_i \geq 0, i = 1..9$

Что бы не описывать большой алгоритм имеющий большое количество действий, которые нам не будут нужны, так как мы уже имеем каноническую задачу. Опишу пример, разобрав пример вы лучше поймёте сам алгоритм.

Решение задачи ЛП симплекс-методом

Целевая функция:

$$10x_1 + 50x_2 + 0 * x_3 + 10x_4 + 70x_5 + 40x_6 + 0 * x_7 + 60x_8 + 0 * x_9 \to min$$

Условия:

$$3\mathrm{x}_1 + 2\mathrm{x}_2 + 2\mathrm{x}_3 + 1\mathrm{x}_4 + 1\mathrm{x}_5 + 1\mathrm{x}_6 + 2\mathrm{x}_7 + 0 * \mathrm{x}_8 + 0 * \mathrm{x}_9 = 60$$

$$0 * \mathrm{x}_1 + 1\mathrm{x}_2 + 0 * \mathrm{x}_3 + 2\mathrm{x}_4 + 0 * \mathrm{x}_5 + 1\mathrm{x}_6 + 0 * \mathrm{x}_7 + 0 * \mathrm{x}_8 + 2\mathrm{x}_9 = 90$$

$$0 * \mathrm{x}_1 + 0 * \mathrm{x}_2 + 3\mathrm{x}_3 + 1\mathrm{x}_4 + 5\mathrm{x}_5 + 3\mathrm{x}_6 + 3\mathrm{x}_7 + 8\mathrm{x}_8 + 4\mathrm{x}_9 = 320$$

Добавим к системе искусственные базисы (это необходимо для создания универсального алгоритма).

$$3\mathrm{x}_1 + 2\mathrm{x}_2 + 2\mathrm{x}_3 + 1\mathrm{x}_4 + 1\mathrm{x}_5 + 1\mathrm{x}_6 + 2\mathrm{x}_7 + 0 * \mathrm{x}_8 + 0 * \mathrm{x}_9 + \mathrm{R1} = 60$$

$$0 * \mathrm{x}_1 + 1\mathrm{x}_2 + 0 * \mathrm{x}_3 + 2\mathrm{x}_4 + 0 * \mathrm{x}_5 + 1\mathrm{x}_6 + 0 * \mathrm{x}_7 + 0 * \mathrm{x}_8 + 2\mathrm{x}_9 + \mathrm{R2} = 90$$

$$0 * \mathrm{x}_1 + 0 * \mathrm{x}_2 + 3\mathrm{x}_3 + 1\mathrm{x}_4 + 5\mathrm{x}_5 + 3\mathrm{x}_6 + 3\mathrm{x}_7 + 8\mathrm{x}_8 + 4\mathrm{x}_9 + \mathrm{R3} = 320$$

То есть:

3X1+2X2+2X3+1X4+1X5+1X6+2X7+0X8+0X9+R1=60

0X1+1X2+0X3+2X4+0X5+1X6+0X7+0X8+2X9+R2=90

0X1+0X2+3X3+1X4+5X5+3X6+3X7+8X8+4X9+R3=320

Переходим к формированию исходной симплекс таблицы. В строку F таблицы заносятся коэффициенты целевой функции.

Так как среди исходного набора условий были равенства, мы ввели искусственные переменные R. Это значит, что в симплекс таблицу нам необходимо добавить дополнительную строку M, элементы которой рассчитываются как сумма соответствующих элементов условий-равенств (тех которые после приведения к каноническому виду содержат искусственные переменные R) взятая с противоположным знаком.

Из данных задачи составляем исходную симплекс таблицу.

	X1	X2	X3	X4	X5	X6	X7	X8	X9	Свободные члены
F	10	50	0	10	70	40	0	60	0	0
R1	3	2	2	1	1	1	2	0	0	60
R2	0	1	0	2	0	1	0	0	2	90
R3	0	0	3	1	5	3	3	8	4	320
M	-3	-3	-5	-4	-6	-5	-5	-8	-6	-470

Так как в столбце свободных членов нет отрицательных элементов, то найдено допустимое решение. В строке M имеются отрицательные элементы, это означает что полученное решение не оптимально. Определим ведущий столбец. Для этого найдем в строке M максимальный по модулю отрицательный элемент - это -8.

Ведущей строкой будет та для которой отношение свободного члена к соответствующему элементу ведущего столбца минимально. Ведущей строкой является R3, а ведущий элемент: 8.

	X1	X2	X3	X4	X5	X6	X7	X9	Свободные члены
F	10	50	-22.5	2.5	32.5	17.5	-22.5	-30	-2400
R1	3	2	2	1	1	1	2	0	60
R2	0	1	0	2	0	1	0	2	90
X8	0	0	0.375	0.125	0.625	0.375	0.375	0.5	40
M	-3	-3	-2	-3	-1	-2	-2	-2	-150

Если не ясно как вычислили таблицу, читать «**Ещё немного теории» на странице 14**

В строке M имеются отрицательные элементы, это означает что полученное решение не оптимально. Определим ведущий столбец. Для этого найдем в строке M максимальный по модулю отрицательный элемент - это -3. Ведущей строкой будет та для которой отношение свободного члена к соответствующему элементу ведущего столбца минимально. Ведущей строкой является R1, а ведущий элемент: 3.

	X2	X3	X4	X5	X6	X7	X9	Свободные члены
F	43.333	-29.167	-0.833	29.167	14.167	-29.167	-30	-2600
X1	0.667	0.667	0.333	0.333	0.333	0.667	0	20
R2	1	0	2	0	1	0	2	90
X8	0	0.375	0.125	0.625	0.375	0.375	0.5	40
M	-1	0	-2	0	-1	0	-2	-90

В строке M имеются отрицательные элементы, это означает что полученное решение не оптимально. Определим ведущий столбец. Для этого найдем в строке M максимальный по модулю отрицательный элемент - это -2. Ведущей строкой будет та для которой отношение свободного члена к соответствующему элементу ведущего столбца минимально. Ведущей строкой является R2, а ведущий элемент: 2.

	X2	X3	X5	X6	X7	X9	Свободные члены
F	43.75	-29.167	29.167	14.583	-29.167	-29.167	-2562.5
X1	0.5	0.667	0.333	0.167	0.667	-0.333	5
X4	0.5	0	0	0.5	0	1	45
X8	-0.063	0.375	0.625	0.313	0.375	0.375	34.375
M	0	0	0	0	0	0	0

В строке M отрицательные элементы отсутствуют. Рассмотрим строку F в которой имеются отрицательные элементы, это означает что полученное решение не оптимально. Определим ведущий столбец. Для этого найдем в строке F максимальный по модулю отрицательный элемент - это -29.167. Ведущей строкой будет та для которой положительное отношение свободного члена к соответствующему элементу ведущего столбца минимально. Ведущей строкой является X1, а ведущий элемент: 0.667.

	X2	X1	X5	X6	X7	X9	Свободные члены
F	65.625	43.75	43.75	21.875	-0	-43.75	-2343.75
X3	0.75	1.5	0.5	0.25	1	-0.5	7.5
X4	0.5	0	0	0.5	0	1	45
X8	-0.344	-0.563	0.438	0.219	0	0.563	31.563
M	0	0	0	0	0	0	0

В строке M отрицательные элементы отсутствуют. Рассмотрим строку F в которой имеются отрицательные элементы, это означает что полученное решение не оптимально. Определим ведущий столбец. Для этого найдем в строке F максимальный по модулю отрицательный элемент - это -43.75 Ведущей строкой будет та для которой положительное отношение свободного члена к соответствующему элементу ведущего столбца минимально. Ведущей строкой является X4, а ведущий элемент: 1.

	X2	X1	X5	X6	X7	X4	Свободные члены
F	87.5	43.75	43.75	43.75	-0	43.75	-375
X3	1	1.5	0.5	0.5	1	0.5	30
X9	0.5	0	0	0.5	0	1	45
X8	-0.625	-0.563	0.438	-0.063	0	-0.563	6.25
M	0	0	0	0	0	0	0

В строке M отрицательные элементы отсутствуют. Рассмотрим строку F в которой имеются отрицательные элементы, это означает что полученное решение не оптимально. Определим ведущий столбец. Для этого найдем в строке F максимальный по модулю отрицательный элемент - это -0 Ведущей строкой будет та для которой положительное отношение свободного члена к соответствующему элементу ведущего столбца минимально. Ведущей строкой является X3, а ведущий элемент: 1.

	X2	X1	X5	X6	X3	X4	Свободные члены
F	87.5	43.75	43.75	43.75	0	43.75	-375
X7	1	1.5	0.5	0.5	1	0.5	30
X9	0.5	0	0	0.5	0	1	45
X8	-0.625	-0.563	0.438	-0.063	0	-0.563	6.25
M	0	0	0	0	0	0	0

Так как в строке F нет отрицательных элементов, то найдено оптимальное решение. Так как исходной задачей был поиск минимума, оптимальное решение есть свободный член строки F, взятый с противоположным знаком. Найдено оптимальное решение F=375 при значениях переменных равных: X7=30, X9=45, X8=6.25, Сравним наш ответ с ответом в книге:

Решением этой задачи является:

$$x_1 = x_2 = x_4 = x_5 = x_6 = x_7 = 0$$

$$x_3 = 30$$

$$x_8 = 6{,}25$$

$$x_9 = 45$$

Тогда необходимо 30+45+6,25=81,25 рулона, а отходы составляют 6,25*60=375 см.

Как видим, всё верно ☺

Следовательно, наша задача была решена верно и алгоритм является полностью подходящим для решения этой задачи ☺

Решения задач подобного типа и другие методы оптимизации, часто встречаются на предприятиях и там их проектирует большое количество специалистов, по этому готовые решения подобных задач найти достаточно сложно. Так как их разрабатывают на предприятиях и они являются производственной тайной. А значит полностью принадлежат компании.

В данном примере вы немного прикоснулись к миру задач линейного программирования и задачам оптимизации. Как видите эти задачи имеют вполне реальное применение ☺

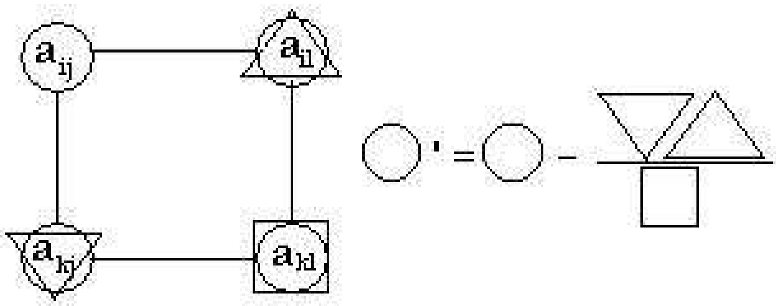
Ещё немного теории

Правила преобразований симплексной таблицы.

При составлении новой симплекс-таблицы в ней происходят следующие изменения:

- Вместо базисной переменной x_k записываем x_l; вместо небазисной переменной x_l записываем x_k.
- ведущий элемент заменяется на обратную величину $a_{k,l}' = 1/a_{k,l}$
- все элементы ведущего столбца (кроме $a_{k,l}$) умножаются на $-1/a_{k,l}$
- все элементы ведущей строки (кроме $a_{k,l}$) умножаются на $1/a_{k,l}$
- оставшиеся элементы симплекс-таблицы преобразуются по формуле $\mathbf{a_{i,j}' = a_{i,j} - (a_{i,l} * a_{k,j} / a_{k,l})}$

Схему преобразования элементов симплекс-таблицы (кроме ведущей строки и ведущего столбца) называют схемой "прямоугольника".



Преобразуемый элемент $\mathbf{a_{i,j}}$ и соответствующие ему три сомножителя как раз и являются вершинами "прямоугольника".

Ещё больше теории тут: http://uchimatchast.ru/teory/simplex.php

Реализация

```
class SimplexMethod
```

Это уже реализованное решение этой задачи, а именно:

Использовать так:

```
double[] minFunction = { 10, 50, 0, 10, 70, 40, 0, 60, 0 };

double[][] constraints = new double[3][];
constraints[0] = new double[] { 3, 2, 2, 1, 1, 1, 2, 0, 0 };
constraints[1] = new double[] { 0, 1, 0, 2, 0, 1, 0, 0, 2 };
constraints[2] = new double[] { 0, 0, 3, 1, 5, 3, 3, 8, 4 };
int[] equations = { 2, 2, 2 }; //EQUAL_TO
// The right-hand sides of the constraints:
double[] rhs = { 60, 90, 320 };

SimplexMethod a = new SimplexMethod();
a.init(minFunction, constraints, equations, rhs);

Tuple<double[], double> xAndMin=a.solveCanonicalToMin(true);

Console.WriteLine("x = {0}, min={1}", string.Join(",",
xAndMin.Item1), xAndMin.Item2);
```

Думаю, всё весьма понятно.

minFunction - Целевая функция

constraints – Условия (но, только левая часть)

equations – всегда массив из двоек, но вы можете реализовать и для < и >, а ещё для ≤ и ≥. Это фактически наши равенства …=60 …=90 …=320. У нас их 3, вот и тут массив из трёх двоек. Почему двоек? Ответ в коде реализации:

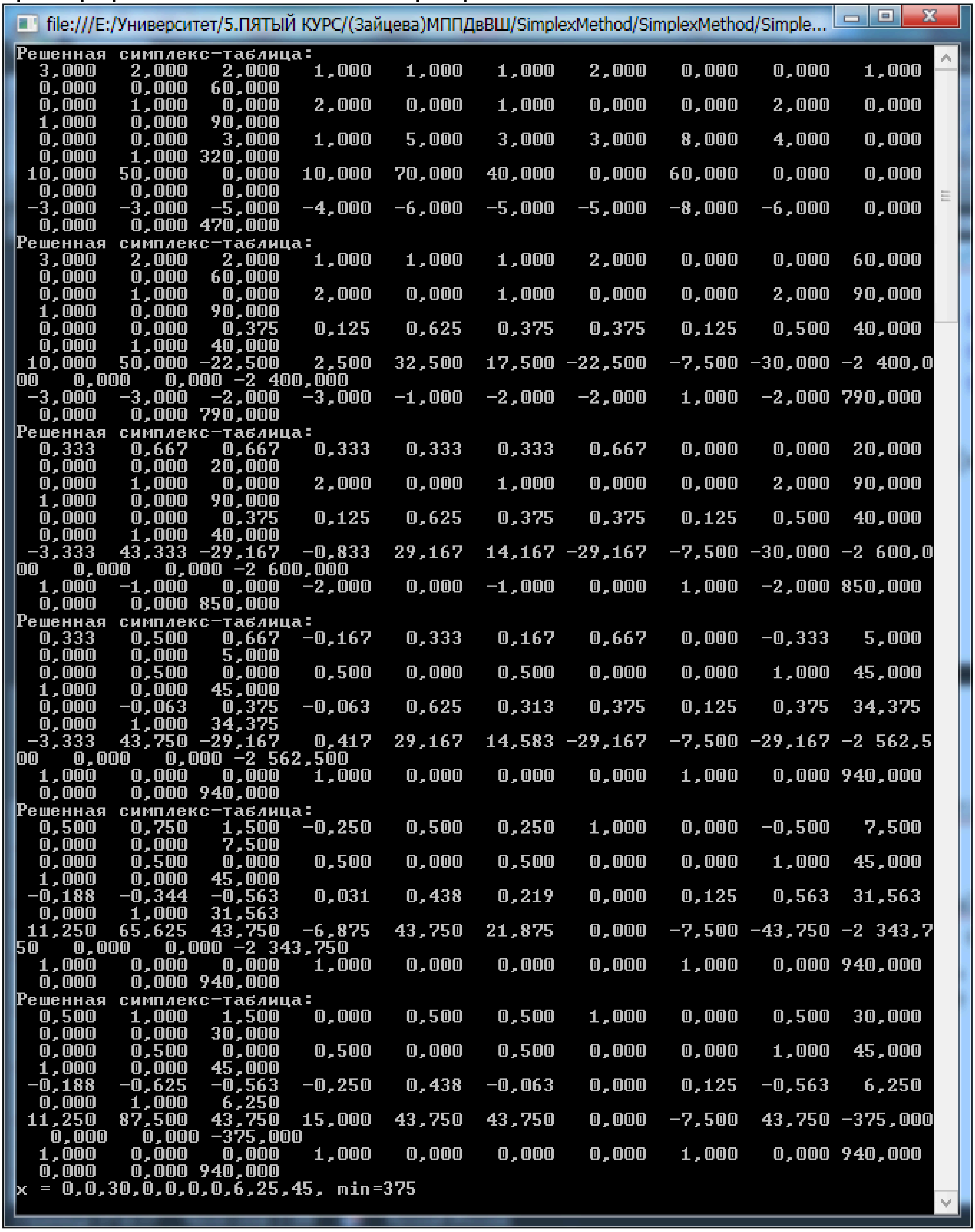
```
class SimplexMethod
    {
        public static int LESS_THAN = 0;
        public static int GREATER_THAN = 1;
        public static int EQUAL_TO = 2;
…
```

rhs – Условия (но, только правая часть).Массив из 60,90,320.

a.init(minFunction, constraints, equations, rhs) – функция инициализации.

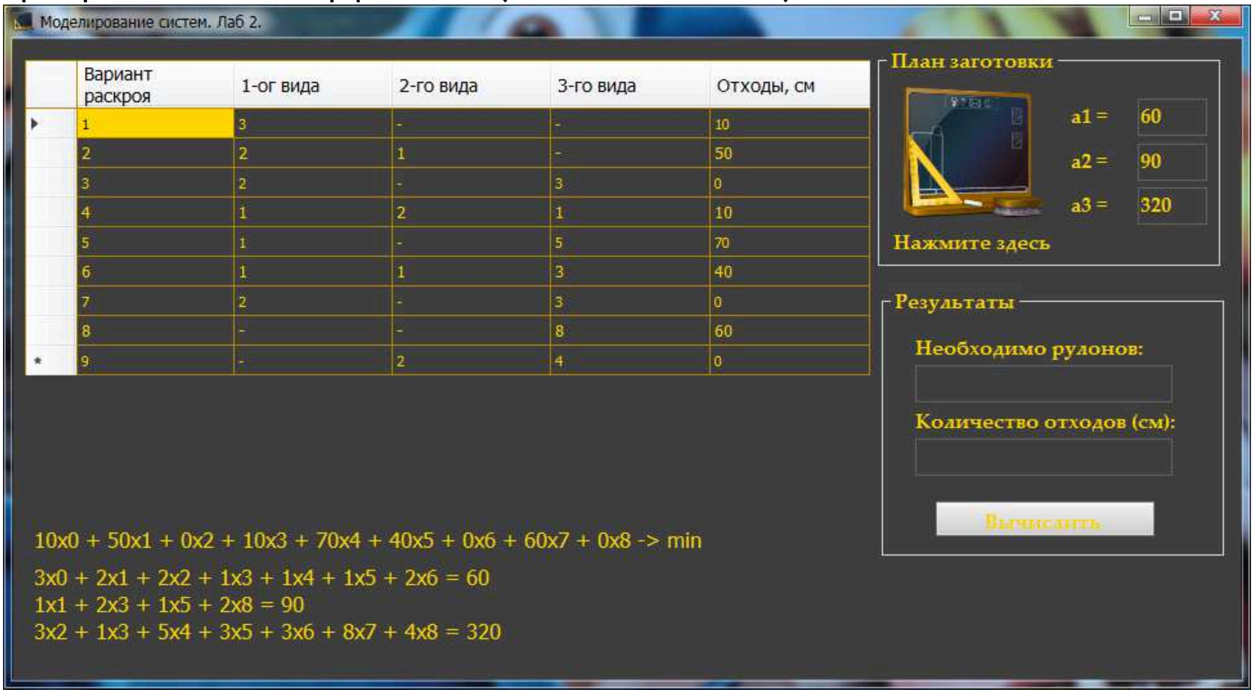
a.solveCanonicalToMin(true) – если передать true, то будут печататься все симплекс таблицы. Сам метод возвращает Tuple<double[], double> - где первый элемент массив значений иксов, а второй элемент минимум функции ☺

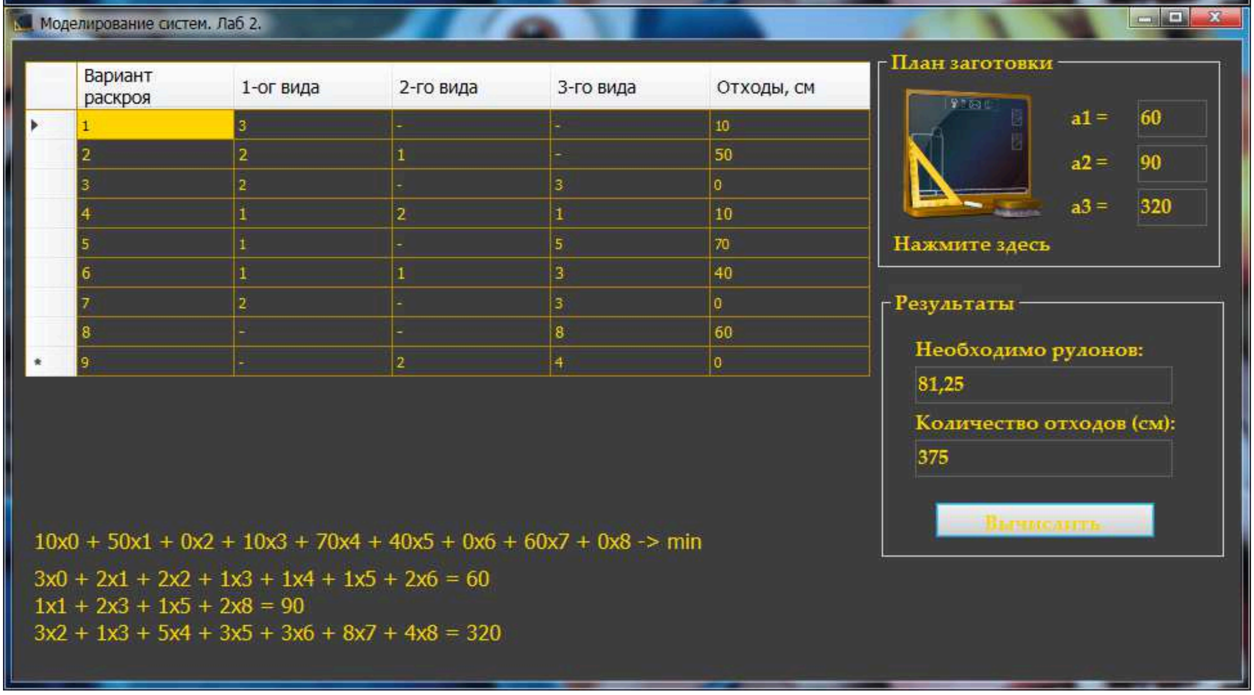
Пример работы консольной программы.




```
Решенная симплекс-таблица:
   3,000    2,000    2,000    1,000    1,000    1,000    2,000    0,000    0,000    1,000
   0,000    0,000   60,000
   0,000    1,000    0,000    2,000    0,000    1,000    0,000    0,000    2,000    0,000
   1,000    0,000   90,000
   0,000    0,000    3,000    1,000    5,000    3,000    3,000    8,000    4,000    0,000
   0,000    1,000  320,000
  10,000   50,000    0,000   10,000   70,000   40,000    0,000   60,000    0,000    0,000
   0,000    0,000    0,000
  -3,000   -3,000   -5,000   -4,000   -6,000   -5,000   -5,000   -8,000   -6,000    0,000
   0,000    0,000  470,000
Решенная симплекс-таблица:
   3,000    2,000    2,000    1,000    1,000    1,000    2,000    0,000    0,000   60,000
   0,000    0,000   60,000
   0,000    1,000    0,000    2,000    0,000    1,000    0,000    0,000    2,000   90,000
   1,000    0,000   90,000
   0,000    0,000    0,375    0,125    0,625    0,375    0,375    0,125    0,500   40,000
   0,000    1,000   40,000
  10,000   50,000  -22,500    2,500   32,500   17,500  -22,500   -7,500  -30,000   -2 400,0
00    0,000    0,000 -2 400,000
  -3,000   -3,000   -2,000   -3,000   -1,000   -2,000   -2,000    1,000   -2,000  790,000
   0,000    0,000  790,000
Решенная симплекс-таблица:
   0,333    0,667    0,667    0,333    0,333    0,333    0,667    0,000    0,000   20,000
   0,000    0,000   20,000
   0,000    1,000    0,000    2,000    0,000    1,000    0,000    0,000    2,000   90,000
   1,000    0,000   90,000
   0,000    0,000    0,375    0,125    0,625    0,375    0,375    0,125    0,500   40,000
   0,000    1,000   40,000
  -3,333   43,333  -29,167   -0,833   29,167   14,167  -29,167   -7,500  -30,000   -2 600,0
00    0,000    0,000 -2 600,000
   1,000   -1,000    0,000   -2,000    0,000   -1,000    0,000    1,000   -2,000  850,000
   0,000    0,000  850,000
Решенная симплекс-таблица:
   0,333    0,500    0,667   -0,167    0,333    0,167    0,667    0,000   -0,333    5,000
   0,000    0,000    5,000
   0,000    0,500    0,000    0,500    0,000    0,500    0,000    0,000    1,000   45,000
   1,000    0,000   45,000
   0,000   -0,063    0,375   -0,063    0,625    0,313    0,375    0,125    0,375   34,375
   0,000    1,000   34,375
  -3,333   43,750  -29,167    0,417   29,167   14,583  -29,167   -7,500  -29,167   -2 562,5
00    0,000    0,000 -2 562,500
   1,000    0,000    0,000    1,000    0,000    0,000    0,000    1,000    0,000  940,000
   0,000    0,000  940,000
Решенная симплекс-таблица:
   0,500    0,750    1,500   -0,250    0,500    0,250    1,000    0,000   -0,500    7,500
   0,000    0,000    7,500
   0,000    0,500    0,000    0,500    0,000    0,500    0,000    0,000    1,000   45,000
   1,000    0,000   45,000
  -0,188   -0,344   -0,563    0,031    0,438    0,219    0,000    0,125    0,563   31,563
   0,000    1,000   31,563
  11,250   65,625   43,750   -6,875   43,750   21,875    0,000   -7,500  -43,750   -2 343,7
50    0,000    0,000 -2 343,750
   1,000    0,000    0,000    1,000    0,000    0,000    0,000    1,000    0,000  940,000
   0,000    0,000  940,000
Решенная симплекс-таблица:
   0,500    1,000    1,500    0,000    0,500    0,500    1,000    0,000    0,500   30,000
   0,000    0,000   30,000
   0,000    0,500    0,000    0,500    0,000    0,500    0,000    0,000    1,000   45,000
   1,000    0,000   45,000
  -0,188   -0,625   -0,563   -0,250    0,438   -0,063    0,000    0,125   -0,563    6,250
   0,000    1,000    6,250
  11,250   87,500   43,750   15,000   43,750   43,750    0,000   -7,500   43,750 -375,000
    0,000    0,000 -375,000
   1,000    0,000    0,000    1,000    0,000    0,000    0,000    1,000    0,000  940,000
   0,000    0,000  940,000
x = 0,0,30,0,0,0,0,6,25,45, min=375
```

Программа с интерфейсом (начинка та же).





Другие задачи.