**Метод ветвей и границ**

*Формат входных данных:*

Обязательные параметры:

а) Функция в явном виде;

б) Ограничения типа равенства или неравенства в явном виде;

*\*количество ограничений и состав переменных в них может варьироваться в зависимости от выбора пользователя*

*Формат выходных данных:*

а) координаты точки экстремума;

б) значение функции в точке экстремума

**Алгоритм на примере конкретной задачи.**

1. **Дано:**

Изображение выглядит как текст

Автоматически созданное описание

1. **Решаем систему уравнений, состоящую из ограничений функции (вместо знаков <=, >= ставим знак =):**

**2х1 + х2 = 10**

**x1+4x2 = 13**

**Таким образом, х1 = 27/7 (или 3.857), х2 = 16/7 (или 2.285). z(x1, x2) = 14.57**

**Для решения системы в питоне:**

import numpy # импортируем библиотеку

M1 = numpy.array([[2., 5.], [1., -10.]]) # Матрица (левая часть системы)

v1 = numpy.array([1., 3.]) # Вектор (правая часть системы)

#Запишем все числа с точкой, т.к. иначе в Python 2 они будут участвовать в целочисленных операциях (остатки от деления будут отбрасываться)

numpy.linalg.solve(M1, v1)

1. **Имеем х1 = 3.857, х2 = 2.285. Находим так сказать окрестности эти точек. У x1 это 3 и 4, у х2 = 2 и 3.**

**Создаем все возможные комбинации из x1 и x2. Получаем: (3; 2), (3; 3), (4; 2), (4;3).**

**Далее проверяем каждую выбранную комбинацию на удовлетворение изначальных ограничений:Изображение выглядит как текст, часы, датчик

Автоматически созданное описание**

**а) Рассматриваем точку (3; 2). 2\*3 + 2 <= 10 – верно. 3 + 4\*2 <= 13 – верно. Таким образом, точка (3; 2) – первый кандитат.**

**б) Рассматриваем точку (3; 3). 2\*3 + 3 <= 10 – верно. 3 + 4\*3 <= 13 – неверно. Таким образом, точка (3; 3) отбрасывается, так как не удовлетворяет одному условию.**

**в) Рассматриваем точку (4; 2). 2\*4 + 2 <= 10 – верно. 4 + 4\*2 <= 13 –верно. Таким образом, точка (4; 2) – еще один кандидат, так как удовлетворяет всем условиям.**

**г) Рассматриваем точку (4; 3). 2\*4 + 3 <= 10 – неверно. 4 + 4\*3 <= 13 – неверно. Таким образом, точка (4; 3) отбрасывается, так как не удовлетворяет двум условиям.**

1. **Таким образом, всем условиям удовлетворили две точки (3;2) и (4;2). Подставляем эти две точки в функцию, которую мы максимизируем и считаем ее значение. В какой точке окажется самое высокое значение функции, ту точку и пишем в ответ.**

****

**А) 2\*3+3\*3 = 15 (точка (3;2))**

**Б) 2\*4 + 3\*2 = 14 (точка (4;2))**

**Таким образом, 15 > 14 => выбираем точку (3;2).**

**Ответ: (3;2); z = 15.**

**Что протестировать:**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Метод Гомори**

*Формат входных данных:*

Обязательные параметры:

а) Функция в явном виде;

б) Ограничения типа равенства или неравенства в явном виде;

*\*количество ограничений и состав переменных в них может варьироваться в зависимости от выбора пользователя*

*Формат выходных данных:*

а) координаты точки экстремума;

**Алгоритм:**

*Шаг 1*. Решается линейно ослабленная задача. Если все переменные – целые, задача решена, если нет – переходим к шагу 2.

*Шаг 2* Убедиться, что ответ получился нецелочисленным. Если он целочисленный, то задача решена;

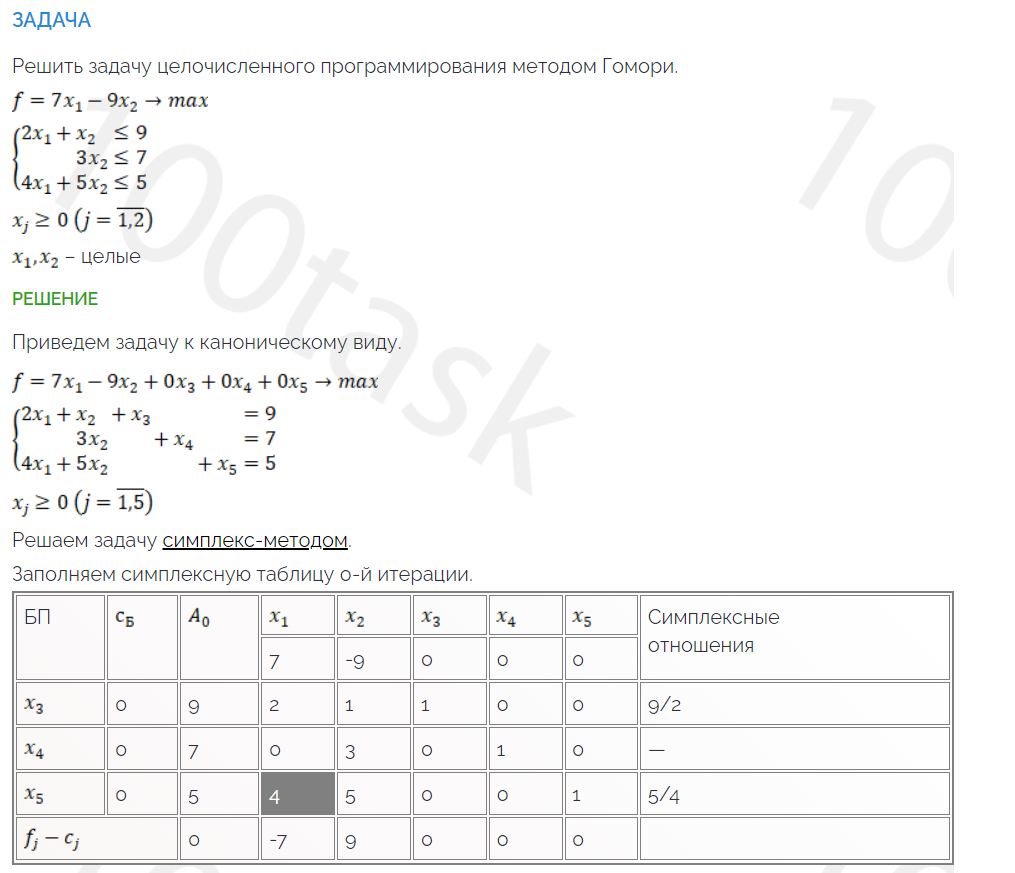
*Шаг 3* Умножить значения в последней строке (строка F) на -1;

*Шаг 4.* В заключительной симплекс-таблице выбирается любое ограничение, у которого правая часть дробная (в столбце *bi*дробное число).

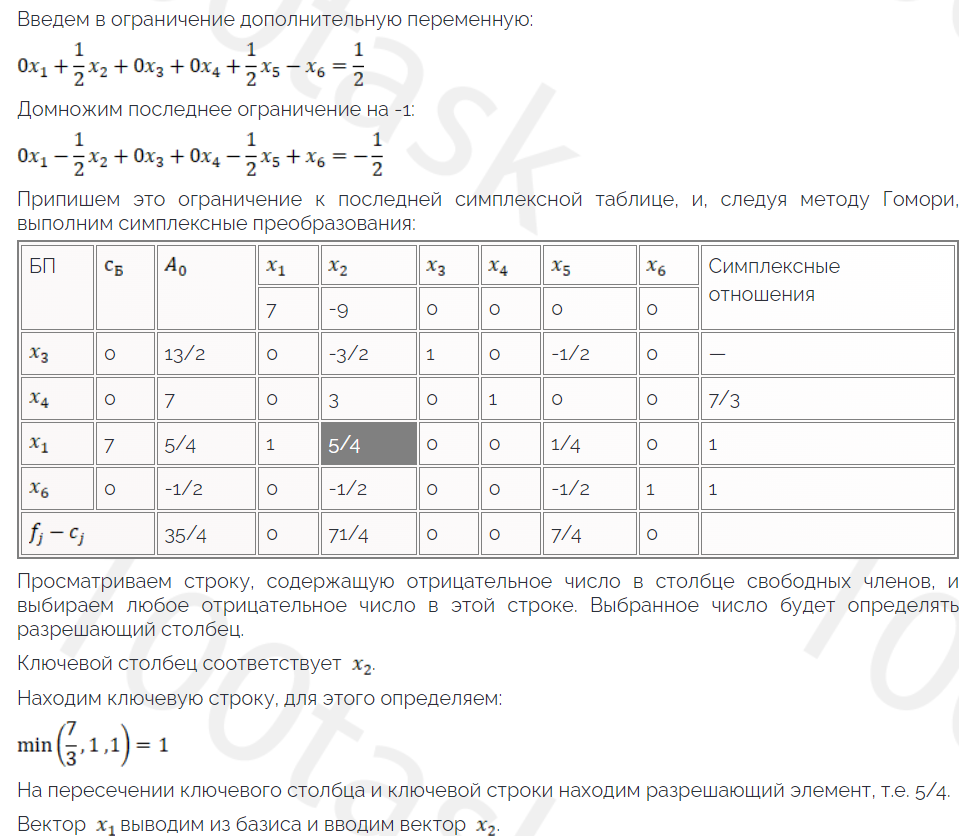
По данному ограничению строится отсечение: соответствующее уравнение записывается таким образом, чтобы все коэффициенты при переменных и правая часть ограничения выглядели в виде суммы целых и дробных частей (причем дробная часть должна быть от нуля до единицы); целые части переносятся в левую часть уравнения, а дробные – в правую. Полученная правая часть и будет уравнением отсечения. (подробнее в примере)

*Шаг 5.* Дополняем последнюю таблицу полученным отсечением и решаем задачу двойственным симплекс-методом. Если вновь получается дробное решение – снова переходим к шагу 4.

**Пример**

****

**Изображение выглядит как стол

Автоматически созданное описание** **** Изображение выглядит как стол

Автоматически созданное описание

**Изображение выглядит как стол

Автоматически созданное описание**

**Еще пример:** [**https://www.matburo.ru/Examples/Files/LP\_Num2.pdf**](https://www.matburo.ru/Examples/Files/LP_Num2.pdf)

**Гиты:** [**https://github.com/tagartagar/simplex**](https://github.com/tagartagar/simplex)

[**https://github.com/dunaevaa/gomory-method**](https://github.com/dunaevaa/gomory-method)

[**https://github.com/dungnv0696/GomoryCut**](https://github.com/dungnv0696/GomoryCut)