Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчет по лабораторной работе №3  
Первая фаза симплекс-метода.

Выполнил:

cтудент гр. 953505

Басенко К. А.

Руководитель:

доцент

Алёхина А. Э.

Минск 2022

Краткие теоретические сведения

Цель первой фазы симплекс метода - выяснить, является ли задача линейного программирования в канонической форме совместной и в случае совместности найти какой-нибудь базисный допустимый план с множеством базисных индексов. Задача распознавания совместности сводится к решению вспомогательной задачи ЛП. Чтобы выяснить, является ли ЗЛП совместной, необходимо:

1. Сделать все свободные члены неотрицательными
2. Составить вспомогательную задачу ЛП. Для того, чтобы составить вспомогательную задачу введем новые искусственные переменные. Искусственных переменных ввести понадобится столько, сколько основных ограничений: по одной искусственной переменной на одно основное ограничение.

Решаем задачу симплекс-методом. Для этого понадобится начальный базисный план. Чтобы получить начальный базисный план надо значения неискусственных переменных положить равными нулю. Значения искусственных переменных однозначно определяются из основных ограничений.

В качестве начального базисного плана берем вектор из нулей, количество которых равно количеству неискусственных переменных, и свободных членов. Этому базисному допустимому плану соответствует множество базисных индексов, которые состоят из индексов искусственных переменных. Получили вспомогательную задачу, которую решаем симплекс-методом.

Если вспомогательная задача совместна, и ее целевая функция ограничена сверху, то с помощью симплекс-метода получим оптимальный базисный план с множеством базисных индексов *В*.

Если в оптимальном плане значения всех искусственных переменных равны 0, то исходная задача ЛП является совместной. Из базисного плана убираем искусственные переменные.

Если в базисном плане значения, соответствующие неискусственным переменным, равны 0, то за множество базисных индексов берем ***B***, если там нет индексов искусственных переменных. В ином случае по алгоритму корректируем множество базисных индексов так, что ***B*** становится текущим множеством базисных индексов.

Алгоритм итерационный: на каждой итерации из множества ***В*** выбирается индекс искусственной переменной, который либо удаляется из ***В***, либо заменяется на индекс неискусственной переменной.

Пусть *В* содержит индекс искусственной переменной , где ***i*** - количество неискусственных переменных. Заменим этот индекс на индекс неискусственной переменной. Для каждого небазисного индекса ***j*** вычислим вектор ***l***:  *= \*.*

Два случая:

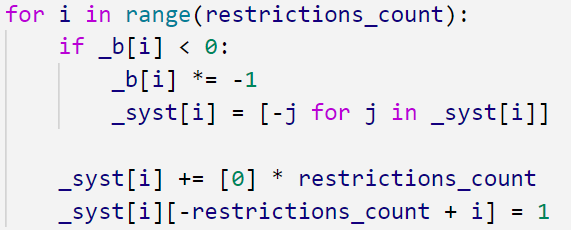
1. Среди небазисных индексов нашли индекс ***j***, для которого . В этом случае в ***В*** заменяем на .
2. Для любого небазисного индекса ***j*** . В этом случае система основных ограничений исходной задачи имеет избыточное ограничение, удаление которого не меняет множество допустимых планов. Таким ограничением является ***i***-е ограничение, которое мы убираем. Из вспомогательной задачи удаляем искусственную переменную с индексом . Из ***В*** удаляем .

Программная реализация

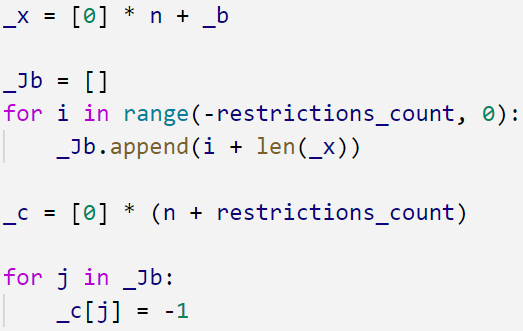
Первая фаза симплекс-метода реализована в функциии, на вход в которую идут: матрица условий *\_syst*, вектор ограничений *b:*



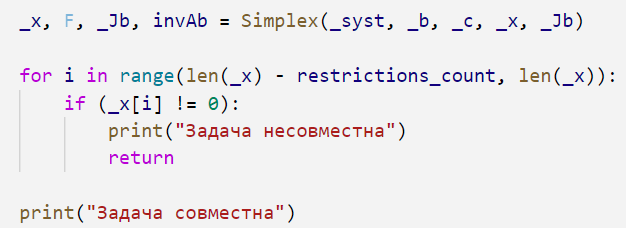
Делаем свободные члены неотрицательными:



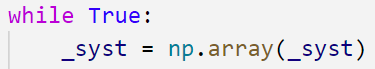
Чтобы найти начальный базисный план, значения неискусственных переменных положим равными 0:



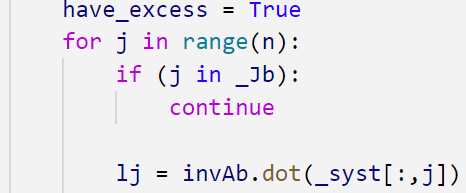
Решаем вспомогательную задачу, проверяем совместность:



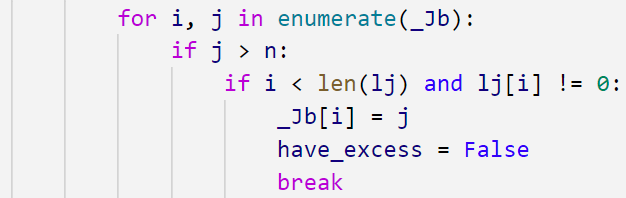
Находим базисные индексы:



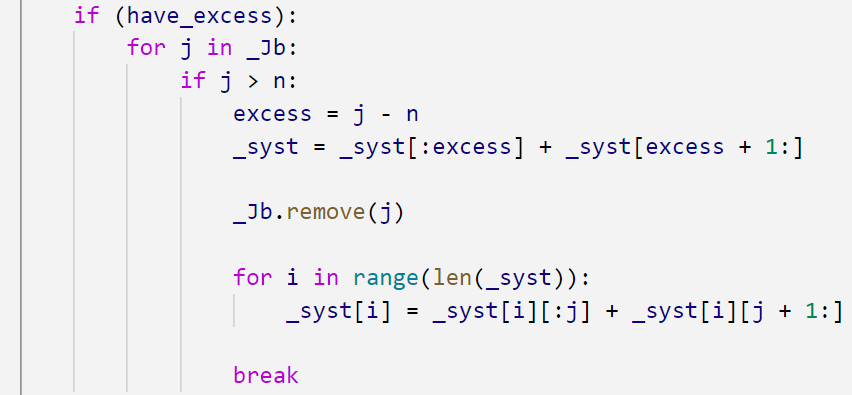
Вычисляем вектор для каждого небазисного индекса:



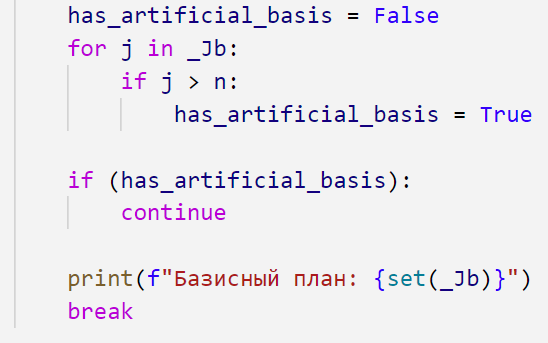
Если среди небазисных индексов нашли индекс ***j***, для которого :



Если есть избыточное ограничение, убираем его; удаляем искусственную переменную из этого ограничения:

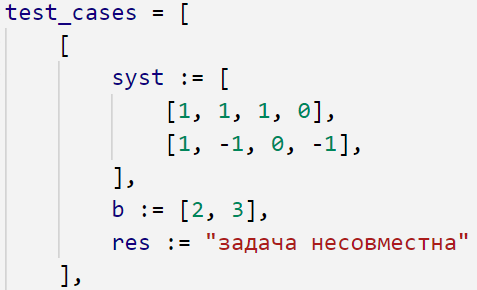


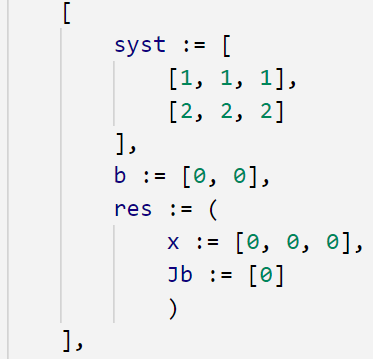
Если в базисном плане значения, соответствующие искусственным переменным, не равны 0, то вычисляем заново базисные индексы:

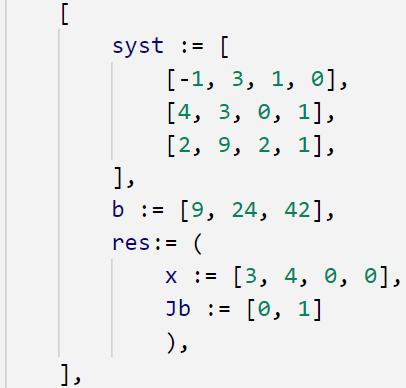


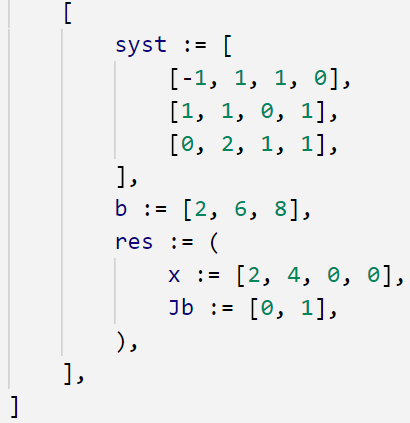
Тестовые примеры

Входные данные для тестирования:

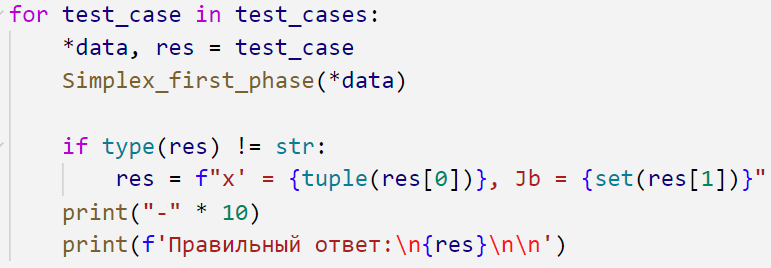








Выполняется метод с предоставленными данными, возвращается результат, выводится полученный результат, выводится ожидаемый результат:

 Вывод, соответствующий предоставленным входным данным:

