Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчет по лабораторной работе №2  
Основная фаза симплекс метода.

Выполнил:

cтудент гр. 953505

Басенко К. А.

Руководитель:

доцент

Алёхина А. Э.

Минск 2022

Краткие теоретические сведения

Теория нам говорит, что базисных допустимых планов конечное число. Более того, если задача совместна, и целевая функция канонической задачи ограничена сверху, то существует оптимальный план, который является базисным.

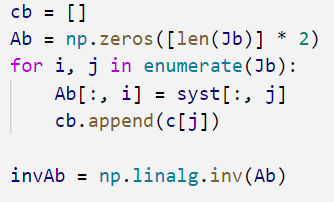
Симплекс-метод перебирает базисные допустимые планы, но делает это систематически. А именно: на первой фазе симплекс-метода находит какой-нибудь базисный допустимый план, либо сообщает, что задача несовместна. Вторая фаза, или основная фаза симплекс-метода, итерационная. На первой итерации симплекс-метод берет найденный на первой фазе базисный допустимый план и по нему строит новый базисный допустимый план, а условием, что значение целевой функции на новом базисном допустимом плане не меньше, чем на старом. Новый базисный допустимый план объявляется текущим. По текущему базисному допустимому плану считается новый базисный допустимый план, для которого значение целевой функции, которое не меньше, чем текущее. И так далее. Процесс останавливается в тот момент, когда симплекс-метод обнаруживает, что достигнут оптимальный базисный план, либо, что целевая функция задачи не ограничена на множестве допустимых планов сверху.

Программная реализация основной фазы симплекс-метода

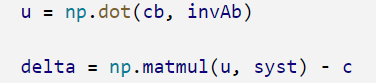
Симплекс-метод реализован в функции, на вход в которую идут: матрица условий *syst*, вектор ограничений *b*, вектор стоимостей *c*, начальный базисный план *x*, базисные индексы *Jb*:



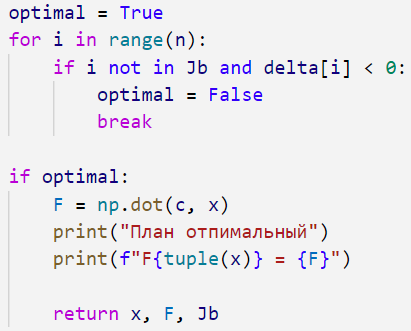
По множеству базисных индексов составляются матрица *AB*, вектор *cB*:



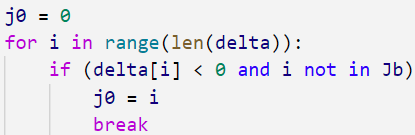
Высчитываются вектор потенциалов и оценок:



Проверяется оптимальность текущего плана:



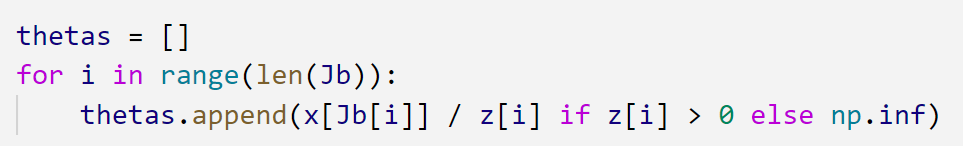
Если план неоптимальный, то среди небазисных отрицательных компонент выберем одну и запомним индекс выбранной компоненты как *j0*:



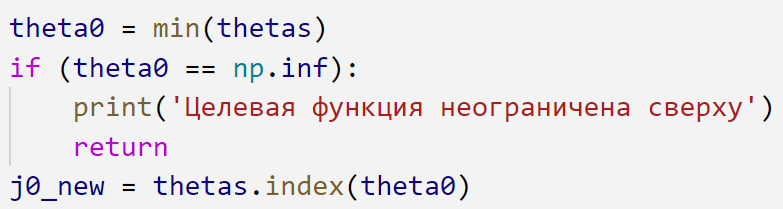
Высчитывается вектор *z =* :



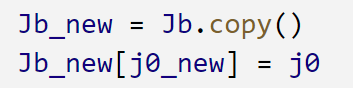
Высчитываются числа *thetas*, количество которых равно количеству базисных индексов:



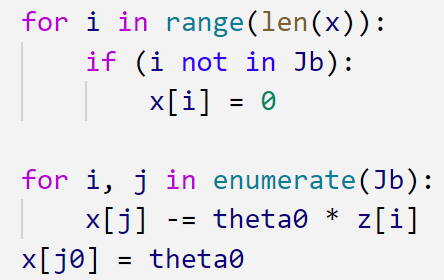
Среди них находим минимальное, запоминаем его индекс:



Далее меняем небазисный индекс *j0* и базисный индекс *j0\_new* местами:



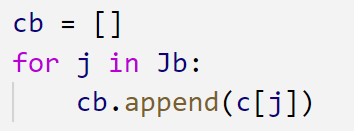
Обновляем компоненты базисного множества:



Задаем базис для следующей итерации:



Для следующей итерации по множеству базисных индексов составляется вектор *cB*:



Для следующей итерации находится матрица, обратная матрице *AB* с заменой столбца:



Далее заново высчитываются вектора потенциалов и оценок, проверяется оптимальность плана и т.д.

Программная реализация обращения матрицы с измененным столбцом

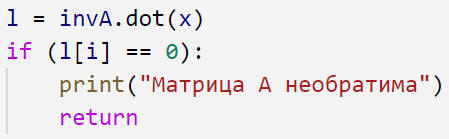
Это действо реализовано в функции, на вход в которую идут: матрица *A*, в которой будет меняться столбец, и обратная которой будет находиться; уже известная матрица, обратная матрице *A --* *invA*; новый столбец *x*, индекс нового столбца *i*:



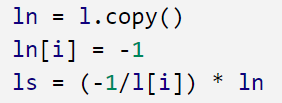
Меняем столбец в матрице *A*:



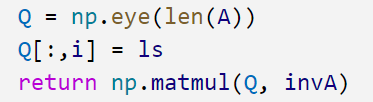
Находим *l* = *invA* ∙ *x*. Если *l* = 0, то матрица *A* необратима и метод завершает свою работу, в противном случае матрица *A* обратима:



Формируем вектор *l* заменой *i*-го элемента на *−1*. Находим *ls*:

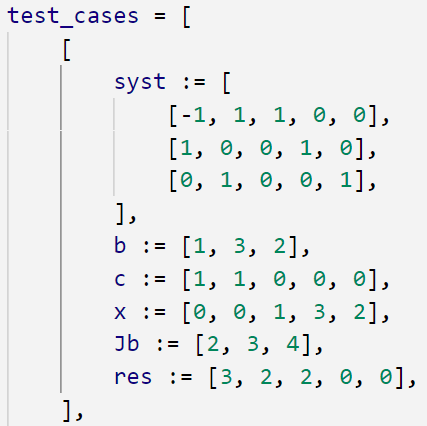


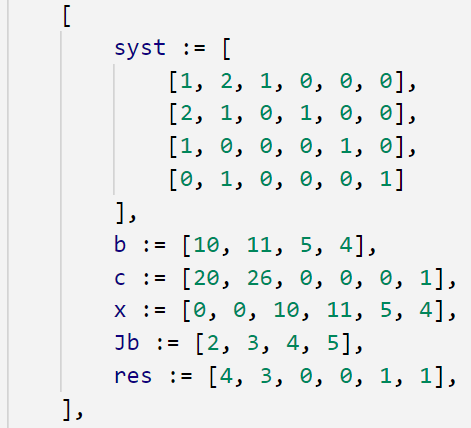
Формируем матрицу *Q*, которая получается из единичной матрицы заменой *i*-го столбца на столбец *ln*, находим обратную матрицу перемножением *Q* и *invA*, возвращаем результат:

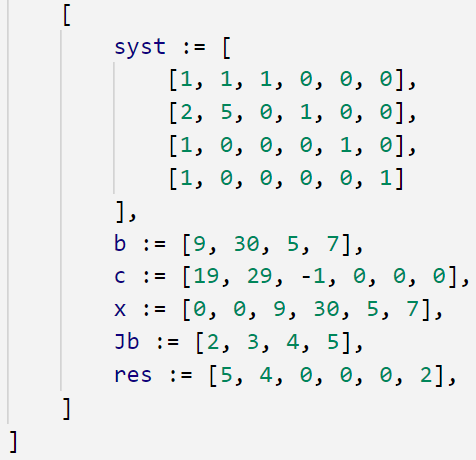


Тестовые примеры

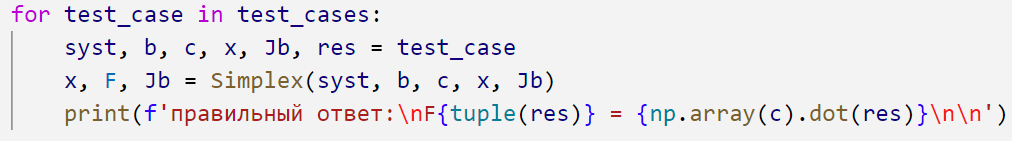
Входные данные для тестирования:







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

 Вывод, соответствующий предоставленным входным данным:

