# СОДЕРЖАНИЕ

Введение

1. Постановка задачи линейного программирования

1.1 Формы задачи линейного программирования

1.2 Переход к канонической форме

2. Симплекс-метод

2.1 Теоретические основы симплекс-метода

2.2 Прямой алгоритм симплексного метода

3. Метод Гомори

4. Математическая и техническая постановка задачи. Программная реализация. Описание проекта

4.1 Запуск

4.2 Описание графического интерфейса

4.3 Описание созданных функций

Заключение

Список литературы

# ВВЕДЕНИЕ

Колоссальные темпы технического прогресса породили проблему создания систем управления сложными системами. Эта проблема приводит к необходимости построения математических моделей принятия оптимальных решений.

Совокупность математических методов, занимающихся вопросами выбора на заданном множестве допустимых решений того решения, которое по установленным критериям является оптимальным, составляет математическую дисциплину «исследование операций».

В свою очередь, исследование операций разделяется на ряд самостоятельных дисциплин, а в данной работе мы столкнемся с задачей решения линейной программы симплексным методом в обычном, целочисленном и частично целочисленном вариантах.

# 1. ПОСТАНОВКА ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ [2]

##

## **1.1 Формы задачи линейного программирования**

В общем виде задача линейного программирования (в дальнейшем ЗЛП) может быть сформулирована как задача нахождения наибольшего значения линейной функции

 (1.1)

на некотором множестве D ⊂ Rn ,где x ∈ D удовлетворяют системе ограничений

 (1.2)

и, возможно, ограничениям

 (1.3)

He умаляя общности, можно считать, что в системе (1.2) первые т ограничений являются неравенствами, а последующие — l-уравнениями. Очевидно, этого всегда можно добиться за счет простого переупорядочения ограничений. Относительно направления знака неравенства будем предполагать, что левая часть меньше или равна правой. Добиться этого можно, умножив на (-1) обе части тех неравенств, которые имеют противоположный знак. Ограничения (1.3), вообще говоря, могут быть рассмотрены как частный случай ограничений в форме неравенств, но в силу особой структуры их обычно выделяют отдельно и называют условиями неотрицательности (или тривиальными ограничениями).

Дополнительно следует заметить, что выбор типа искомого экстремума (максимума или минимума) также носит относительный характер. Так, задача поиска максимума функции

 (1.4)

эквивалентна задаче поиска минимума функции

 (1.5)

Часто условия задачи (1.1) - (1.3), содержащей ограничения только типа неравенств, бывает удобно записывать в сокращенной матричной форме

 (1.6)

где с и x — векторы из пространства Rn, b — вектор из пространства Rm, a А — матрица размерности m × п.

Задачу линейного программирования, записанную в форме (1.1) - (1.3), называют общей задачей линейного программирования (ОЗЛП).

Если все ограничения в задаче линейного программирования являются уравнениями и на все переменные xj наложены условия неотрицательности, то она называется задачей линейного программирования в канонической форме, или канонической задачей линейного программирования (КЗЛП). В матричной форме КЗЛП можно записать в следующем виде:

 (1.7)

Поскольку любая оптимизационная задача однозначно определяется целевой функцией f и областью D, на которой отыскивается оптимум (максимум), будем обозначать эту задачу парой (D, f).

Условимся относительно терминологии, которая используется в дальнейшем и является общепринятой в теории линейного программирования.

Планом ЗЛП называется всякий вектор х из пространства Rn.

**Допустимым планом** называется такой план ЗЛП, который удовлетворяет ограничениям (1.2)-(1.3), т. е. содержится в области D. Сама область D называется при этом областью допустимых планов. **Оптимальным планом** х\* называется такой допустимый план, при котором целевая функция достигает оптимального (в нашем случае — максимального) значения, т. е. план, удовлетворяющий условию

max f(x) = f(x\*).

Величина f\* = f(x\*) называется оптимальным значением целевой функции.

Решением задачи называется пара (х\*, f\*), состоящая из оптимального плана и оптимального значения целевой функции, а процесс решения заключается в отыскании множества всех решений ЗЛП.

## **1.2 Переход к канонической форме**

Подавляющее большинство известных методов решения задач линейного программирования предназначены для канонических задач. Поэтому начальный этап решения всякой общей задачи линейного программирования обычно связан с приведением ее к некоторой эквивалентной канонической задаче.

Общая идея перехода от ОЗЛП к КЗЛП достаточно проста:

* ограничения в виде неравенств преобразуются в уравнения за счет добавления фиктивных неотрицательных переменных , которые одновременно входят в целевую функцию с коэффициентом 0, т. е. не оказывают влияния на ее значение;

* переменные, на которые не наложено условие неотрицательности, представляются в виде разности двух новых неотрицательных переменных:

* переменные, на которые наложено условие неположительности, представляются в виде новой неотрицательной переменной помноженной на -1:

Нетрудно заметить, что «платой» за переход от общей формы задачи линейного программирования к канонической является рост ее размерности, что, при прочих равных условиях, является фактором, усложняющим процесс решения.

# 2. СИМПЛЕКС-МЕТОД

##

## **2.1 Теоретические основы симплекс-метода**

Исходя из свойств линейных экстремальных задач, можно заключить, что на принципиальном уровне поиск их решений сводится к последовательному перебору угловых точек множества допустимых планов или, что то же самое, перебору соответствующих допустимых базисных планов. Средством решения данной проблемы явились прикладные оптимизационные методы, основанные на последовательном, целенаправленном переборе базисных планов ЗЛП.

Классическим методом решения ЗЛП стал симплекс-метод, получивший также в литературе название метода последовательного улучшения плана (упорядоченность обеспечивается монотонным изменением значения целевой функции при переходе к очередному плану), разработанный в 1947 г. американским математиком Джорджем Данцигом.

Пусть стоит задача максимизации

 (2.1)

при условиях

, (2.2)

Xj≥ 0, j=1,…,n (2.3)

Предположим, что нам удалось найти опорный план X0, в котором, например, первые m компонент отличны от нуля:

X0=(X10,X20,…,Xm0, 0, …, 0), (2.4)

и соответствующий базис Б=(A1,A2,…,Am).

Попытаемся выбрать другую систему базисных векторов с целью построения нового опорного плана, в котором k-я переменная (k>m) принимает значениеΘ >0:

X(Θ) = (X1(Θ), X2(Θ),…,Xm(Θ), 0, …,Θ, … 0) (2.5)

Подставляя (2.4) в (2.2), имеем

(2.6)

Подставив (2.5) в (2.2), получаем

 (2.7)

Разложим вектор Ak по векторам исходного базиса

 (8)

В общем случае для получения коэффициентов такого разложения придется решать систему mуравнений с m неизвестными, которая имеет единственное решение, поскольку базисные векторы линейно независимы и соответствующая матрица имеет ненулевой определитель. Заметим, что в ситуации, когда базисные векторы являются единичными (образуют единичную матрицу), искомые коэффициенты совпадают с компонентами исходного вектора; поэтому в дальнейшем мы предпочтем работать с единичным базисом.

Подставляя (2.6) и (2.8) в (2.7), получаем

, (2.9)

откуда имеем

 (2.10)

Так как система уравнений (2.10) имеет единственное решение, то получаем представление первых mкомпонент нового плана

 (2.11)

Естественно потребовать неотрицательность компонент нового плана. Так как нарушение неотрицательности в (2.11) может возникнуть лишь при Zjk>0, то значение Θ нужно взять не превышающим наименьшего из отношений к положительным Zjk.

Если к тому же учесть, что число положительных (базисных) компонент опорного плана должно оставаться равным m, то одну из первых m (ненулевых) компонент исходного плана обращаем в нуль выбором

 (2.12)

Подставляя (2.11) в (2.1), имеем

 (2.13)

Если обозначить

, (2.14)

, (2.15)

то (2.13) примет вид

 (16)

Из полученных соотношений напрашиваются следующие выводы.

Критерий 1 (критерий оптимальности).Если все Δk ≥ 0, то выбранный план для задачи максимизации является оптимальным.

Критерий 2. Если обнаруживается некоторое Δk < 0 и хотя бы одно из значений Zjk >0, то переход к новому плану увеличит значение целевой функции.

Этот вывод с очевидностью следует из (2.16); в такой ситуации согласно (2.12) полагаем k-ю переменную равной Θ и преобразуем значения остальных (базисных) переменных в соответствии с (2.11).

Критерий 3**.** Если обнаруживается некоторое Δk < 0, но все Zjk≤0, то линейная форма задачи не ограничена по максимуму.

Этот вывод следует из того, что согласно (2.11) компоненты нового плана сохраняют неотрицательность при любом Θ>0 (в том числе и при сколь угодно большом) и согласно (2.16) появляется возможность неограниченного изменения значения целевой функции.

Предположение о том, что базисными являются первые mкомпонент плана, не является принципиальным, и указание диапазона поj от 1 до m в (2.11)-(2.15) можно заменить на указание о принадлежности к базису “j∈Б“.

Если все опорные планы задачи являются невырожденными (число положительных компонент равно m), то Θ отлично от нуля и переход к новому плану согласно (2.16) изменяет значение целевой функции, что гарантирует достижение экстремума за конечное число шагов. При наличии вырожденных планов возможно т. н. зацикливание (возврат к ранее рассмотренным планам), но на практике зацикливание никогда не возникало.

## **2.2 Прямой алгоритм симплексного метода [1]**

Пусть исходная задача приведена к канонической форме и начальный базис образует единичную матрицу. Тогда базисные компоненты опорного плана совпадают с правыми частями ограничений и коэффициенты Zjk разложения вектора Xk по такому базису совпадают с компонентами этого вектора.

Для единообразия описания вычислительной процедуры в дальнейшем будем пользоваться т.н. симплексной таблицей вида:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **C** | Базис | **План** | C1 | C2 |  | Cm | Cm+1 |  | Ck |  | Cn |
| **баз** | **плана** | **X** | X1 | X2 |  | Xm | Xm+1 |  | Xk |  | Xn |
| С1 | X1 | B1 | 1 | 0 |  | 0 | Z1m+1 |  | Z1k |  | Z1n |
| С2 | X2 | B2 | 0 | 1 |  | 0 | Z2m+1 |  | Z2k |  | Z2n |
|  |  |  |  |  |  |  |  |  |  |  |  |
| Cm | Xm | Bm | 0 | 0 |  | 1 | Zmm+1 |  | Zmk |  | Zmn |
| Zk | **L(X)** | Z1 | Z2 |  | Zm | Zm+1 |  | Zk |  | Zn |
| **Δk** | Δ1 | Δ2 |  | Δ m | Δ m+1 |  | Δk |  | Δn |

В центральной части таблицы записываются коэффициенты при неизвестных в ограничениях, в столбцеX - правая часть ограничений (базисные компоненты плана), в первой строке - коэффициенты линейной формы, во второй строке – переменные, входящие в целевую функцию и систему ограничений. Основное поле симплекс таблицы - коэффициенты при неизвестных в ограничениях. В первом столбце для удобства вычислений будем заносить коэффициенты линейной формы при базисных переменных, указанных во втором столбце (умножение его на столбец X (свободные члены Bi≥0) с суммированием дает значение L(X); аналогичное умножение его на столбец Xk даст Zk). Последняя строка получается вычитанием из предыдущей строки элементов первой строки таблицы и позволяет судить об оптимальности плана.

Т.к. выбор типа искомого экстремума (максимума или минимума) носит относительный характер, то при решении задач максимизации/минимизации в последней строке должны быть только неотрицательные элементы.

Обратим внимание на определение начального опорного плана. Пусть задача приведена к канонической форме и компоненты вектора правой части неотрицательны. Если в системе векторов коэффициентов при переменных (матрице А) обнаруживается подсистема, образующая единичную подматрицу, то эти векторы образуют базис опорного плана и вектор правой части определяет базисные компоненты этого плана.

Если такой единичной подматрицы не обнаруживается, то либо придется перебирать все подсистемы m уравнений с m неизвестными в надежде обнаружить неотрицательные решения, либо прибегнуть к методу искусственного базиса.

В последнем случае в ограничения добавляют неотрицательные, т.н. искусственные переменные так, чтобы возникла единичная подматрица коэффициентов, и эти переменные включают в линейную форму с коэффициентом - М для задачи максимизации, где М**>**0 **-** сколь угодно большое число.

Полученная М-задача решается до получения оптимального плана.

Если в оптимальном плане М-задачи значения искусственных переменных равны нулю, то значения остальных компонент образуют оптимальный план исходной задачи**.**

Если в оптимальном плане М-задачи значение хотя бы одной из искусственных переменных отлично от нуля, то исходная задача не имеет ни одного плана (ее ограничения противоречивы).

Если некоторая задача решается прямым алгоритмом симплексного метода, то решение сопряженной задачи можно видеть в строке Z конечной симплексной таблицы в позициях, соответствующих начальному единичному базису.

# 3. МЕТОД ГОМОРИ [1]

При решении многих задач (планирование мелкосерийного производства, распределение кораблей по путям сообщения, выработка суждений типа "да-нет" и т.п.) нецелочисленное решение не имеет смысла. Попытка тривиального округления до целых значений приводит либо к нарушению ограничений задачи, либо к недоиспользованию ресурсов. Как мы имели возможность убедиться, для произвольной линейной программы (за исключением программ типа классической транспортной задачи, где коэффициенты матрицы ограничений равны 1 или 0) гарантировать целочисленность решения невозможно.

В случае двухмерной задачи проблема решается относительно просто путем выявления всех целочисленных точек, близких к границе множества планов, построения выпуклого множества планов, содержащего все целочисленные планы и решения задачи над этим множеством.

B общем случае выдвигается идея последовательного отсечения нецелочисленных оптимальных планов**:** обычным симплексным методом отыскивается оптимальный план и, если он нецелочисленный, строится дополнительное ограничение, отсекающее найденный оптимальный план, но не отсекающее ни одного целочисленного плана.

Эта идея, принадлежащая Д. Данцигу и Р. Гомори, впервые была представлена в форме дополнительного ограничения:

 (3.1)

(сумма небазисных компонент оптимального плана должна быть отлична от нуля; хотя бы одна из небазисных компонент должна быть ненулевой). В самом деле, оптимальный план с нулевыми значениями небазисных компонент этому условию не удовлетворяет, что подтверждает отсечение этого плана от исходного множества.

К сожалению, для абсолютного большинства задач скорость сходимости процесса таких отсечений мала. Потому Р. Гомори предложена другая форма дополнительного ограничения. Так, если компонента плана, определяемая k-м уравнением системы ограничений, нецелочисленна, то добавляется ограничение

, (3.2)

где fk - дробная часть компоненты плана (правой части ограничения) и fkj - дробная часть коэффициента при Xj (целая часть числа – наибольшее целое, не превышающее это число; дробная часть числа равна разности между числом и его целой частью), S**\*** - новая дополнительная переменная.

Можно уменьшить объем преобразований, если руководствоваться следующими правилами:

1) выбирать в качестве базового для построения дополнительного ограничения уравнение, определяющее компоненту плана с наибольшей дробной частью;

2) для ввода в базис опорного плана расширенной задачи выбирать переменную, для которой достигается минимум из отношений абсолютных значений Δ j к значениям fk j ;

3) если одна из ранее введенных дополнительных переменных вошла в базис, ее и соответствующее ей уравнение можно отбросить (эта ситуация связана с появлением более жесткого условия, перекрывающего действие ранее введенного).

Появление дополнительного ограничения и дополнительной переменной вновь приводит к проблеме выбора начального опорного плана расширенной задачи и к использованию с этой целью искусственной переменной. Следует заметить, что если при поиске переменной, исключаемой из базиса, значение Θ (определяемое с учетом дополнительного ограничения) соответствует этому ограничению, то можно отказаться от использования искусственной переменной (она все равно выведется из базиса на этом же шаге решения).

Заметим, что для целочисленных программ может обнаружиться отсутствие целочисленных планов (противоречивость ограничений).

Для предложенного здесь метода доказана конечность процесса отсечений, но число этих отсечений непредсказуемо (вполне может обнаружиться быстрое решение задач с десятками переменных и ограничений и фантастически длительное для задач небольших размеров).

#

# 4. МАТЕМАТИЧЕСКАЯ И ТЕХНИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

Математическую и техническую постановку задачи можно сформулировать следующим образом. Разработать программное обеспечение на встроенном языке среды MatLAB, позволяющее решать линейные программы симплексным методом с учетом приведенного выше теоретического материала.

Несмотря на то, что в поставляемом вместе с MatLAB пакете программ Optimization Toolbox имелась функция linprog реализующая решение линейных задач, было принято решение реализовывать симплекс-метод и метод Гомори не используя уже готовые решения, но максимально задействуя встроенные функции среды разработки.

В ходе разработки наибольшее внимание уделялось удобству работы с программой и качеству реализации методов.

В плане ограничений накладываемых на пользователя можно отметить лиши разумность вводимых данных.

# Программная реализация

Программа написана на встроенном в MatLAB языке программирования. Не смотря на всю простоту языка программирования MatLAB’а стоит отметить его большую функциональность обеспечиваемую встроенными функциями и пакетами расширений Toolbox.

Отличительной особенностью разработанной программы является ее графический интерфейс (Рисунок 1), обеспечивающий максимально удобную работы и позволяющий работать с программой даже не посвященным в программирование, математику и экономику людям.

Основная часть программного кода в соответствии со своим назначением разделена на функции, которые хранятся отдельно от интерфейса и могут быть использованы пользователем по своему усмотрению, в том числе в других приложениях/программах или же могут быть вызваны через консоль MatLAB.

Рисунок 1. Графический интерфейс программы

# Описание проекта

##

## **4.1 Запуск**

Для запуска проекта необходимо запустить среду MatLAB и указать путь к каталогу с программой. Затем необходимо запустить графический интерфейс (Рисунок 1) для чего в консоли MatLAB’а требуется ввести guide и в открывшемся окошке (Рисунок 2) выбрать вкладку «Открыть существующий GUI» и указав путь к GUI-интерфейсу «MainSimplexForm.fig» нажать кнопку «ОК».

Рисунок 2. GUIDE - среда разработки и работы с GUI-интерфейсами MatLAB

В итоге появится макет интерфейса (Рисунок 3) для запуска которого достаточно нажать комбинацию клавиш «Ctrl+T» или зеленую стрелочку на панели под главным меню. После запуска перед вами появится полноценный графический интерфейс (Рисунок 1).

Рисунок 3. Макет GUI-интерфейса

##

## **4.2 Описание графического интерфейса**

Вверху формы расположено главное меню (Рисунок 4), состоящее из пунктов «Примеры», «Дополнительно» и «Выход».

Рисунок 4. Главное меню

Пункт «Примеры» позволяет загрузить или создать пример работы с программой. Примеры хранятся в текстовых файлах с расширением «matex» содержащих целевую функцию, ограничения и условия неотрицательности.

Пункт «Дополнительно» позволяет пользователям операционной системы Windows получить доступ к Калькулятору, Блокноту и некоторым другим приложениям.

Пункт «Выход» закрывает среду MatLAB.

Для ввода функции, ограничений, условий неотрицательности и выбора метода используется центральная часть формы (Рисунок 5).

Рисунок 5. Здесь можно ввести исходные данные и выбрать метод решения

Для запуска метода и просмотра результатов его работы служит нижняя часть формы (Рисунок 6).

Рисунок 6. Здесь можно запустить метод и просмотреть результат его работы

Особое внимание надо обратить на статусное сообщение. Поля «Оптимальный план» и «Экстремум функции» будут заполнены последними значениями, но судить об их верности можно только прочитав статусное сообщение! Решение двойственной задачи отображается только для обычного варианта симплексного метода.

В случае если вы хотите воспользоваться частично целочисленным методом стоит обратить внимание на ввод требуемых иксов (Рисунок 7).

Рисунок 7. Ввод номеров требуемых целых иксов

Необходимо через пробел вводить номера иксов, которые вы хотите видеть целыми.

## **4.3 Описание созданных функций**

В ходе разработки проекта были написаны следующие функции:

1. parser\_input – функция разделения строки типа “2x1+5x2-7x3” на массив номеров иксов и массив коэффициентов при этих иксах;
2. parser\_allogr – функция канонизации, формирующая составляющие симплексной таблицы из введенных функции, ограничений, условий неотрицательности;
3. load\_example – функция чтения файла с примером;
4. save\_example – функция записи файла с примером;
5. simple\_simplex – функция, реализующая простой симплекс-метод;
6. gomory\_simplex – функция, реализующая метод Гомори;
7. MainSimplexForm – функция, связанная с графическим интерфейсом и содержащая основные вызовы остальных функций.

Все перечисленные выше функции снабжены подробной справкой и для более подробного ознакомления с ними достаточно в консоли MatLAB ввести: help имя\_функции

Данные функции можно использовать в других программах написанных для MatLAB, т.к. они работают в проекте как «автономные модули».

# Заключение

Итогом написания данной курсовой работы является компьютерная реализация решения линейных программ симплексным методом в обычном, целочисленном и частично целочисленном вариантах.

Разработанная программа позволит упростить решение подобных задач непосвященными во все тонкости пользователям, а также послужит примером для интересующихся экономико-математическими методами людей.

Следует отметить, что симплексный метод в различных вариациях фигурирует во многих задачах дисциплины «исследование операций», например, при рассмотрении двойственности в линейном программировании, целочисленном линейном программировании, матричных играх и т.д. Соответственно созданная программа позволит упростить решение упомянутых выше задач.

# Список литературы

1. Тынкевич М.А. «Экономико-математические методы (исследование операций)», издание 2-е, исправленное и дополненное, Кемерово, 2007г.
2. Конюховский П.В. «Математические методы исследования операций в экономике», СПб.: Питер, 2008г.
3. Дегтярев Ю.И. «Исследование операций», М.: Высшая школа, 2009 г.
4. Ануфриев И.Е., Смирнов А.Б., Смирнова Е.Н. «MATLAB 7», СПб.: БХВ-Петерберг, 2005г.
5. Дъяконов В. «MATLAB 6: Учебный курс», СПб.: Питер, 2010г.