МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

НУКУССКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ИМЕНИ МУХАММАДА АЛ-ХОРЕЗМИ

Факультет «КОМПЬЮТЕРНЫЙ ИНЖИНИРИНГ»

КАФЕДРА информационных технологий

Направления компьютер инжиниринг (ИТ-сервис)

Допустить к защите
Заведующий кафедрой
Айтмуратов Б. Ш.
2017 г. «___»____________

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему: «Решение многокритериальной задачи электроэнергетики в среде MatLab»

Выпускник: **Закаряев И. Ж.**

Научный руководитель: **Мнажов Б.**

НУКУС-2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

При рассмотрении задач исследования операций мы всегда имеем дело с количественной информацией. Но так бывает не всегда: выбор профессии, места работы, проектов научных исследований и т. д. — примеры ситуаций, когда важными являются многие качественные факторы. К этому добавляется неопределенность в исходной информации, связях факторов, последствий нашего выбора, многокритериальность оценивания альтернатив.

Так, часто встречающееся выражение «достичь максимального эффекта при наименьших затратах» уже означает принятие решения при двух критериях. Оценка деятельности предприятий и планирования как системы принятия решений производится на основе более десятка критериев: выполнение плана производства по объему, по номенклатуре, плана реализации, прибыли по показателям рентабельности, производительности труда и т. д.

В задачах математического программирования с одним критерием нужно определить значение целевой функция, соответствующее, например, минимальным затратам или максимальной прибыли. Однако, немного подумав, мы практически в любой реальной ситуации обнаружим несколько целей, противоречащих друг другу. Покажем, насколько широк диапазон проблем, которые могут быть адекватно сформулированы как многокритериальные, и какие характеристики следует использовать в качестве критериев.

Таким образом, для эффективного решения любой из данных задач необходимо в первую очередь построить многокритериальную математическую модель, которую затем нужно оптимизировать, предварительно выбрав наиболее подходящий для этого метод.

Стремление к оптимальному решению - естественное состояние человека, который должен экономить запасы ресурсов (финансовых, энергетических, сырьевых) и времени. Естественное поведение человека - это, как правило, его действия для получения оптимального результата.

Целью данной выпускной квалификационной работы являются получение будущими специалистами основ знаний, необходимых для решения многокритериальных задач в области электроэнергетики.

Выпускная квалификационная работа состоит из введения, 3 глав, заключения, списка использованной литературы и приложения.

В 1 главе рассматриваются общие сведения об электросети и анализ задачи электроэнергетики.

Во 2 главе рассматриваются методы решения многокритериальные оптимизационные задачи электроэнергетики.

В 3 главе рассматриваются задачи электроэнергетики в среде Mat lab и основы работы в среде Mat lab.

В заключении дается общий вывод по проделанной работе. В приложении приводится исходный текст программы.

ГЛАВА 1. ОСНАВНЫЕ СВЕДЕНИЕ ЗАДАЧИ ЭЛЕКТРОЭНЕРГЕТИКИ

1.1. Как устроена электрическая сеть

Передача и распределение электроэнергии осуществляется с помощью воздушных линий электропередачи, пересекающих всю страну. Для уменьшения потерь при передаче электроэнергии в линиях электропередач применяется очень высокое напряжение - десятки и (чаще) сотни киловольт.

В силу своей экономичности при передаче энергии применяется изобретенная русским инженером М.О. Доливо-Добровольским трехфазная система переменного тока, при которой электроэнергия передается с помощью четырех проводов. Три из этих проводов называются линейными или фазными, а четвертый - нейтральным проводом или просто нейтралью.

Потребители электроэнергии рассчитаны на более низкие напряжения, чем напряжение в энергосистеме. Понижение напряжения производится в два этапа. Сначала на понижающей подстанции, являющейся частью энергосистемы, напряжение понижается до 6-10 кВ (киловольт). Дальнейшее понижение напряжение производится на трансформаторных подстанциях. Их знакомые всем стандартные "трансформаторные будки" во множестве разбросаны по предприятиям и жилым массивам. После трансформаторной подстанции напряжение понижается до 220-380 В.

Напряжение между линейными проводами трехфазной системы переменного тока называется линейным.

Источником тока для энергосистемы являются трехфазные генераторы переменного тока, установленные на электростанциях. Каждая из обмоток генератора индуцирует линейное напряжение. Обмотки симметрично расположены по окружности генератора. Соответственно и линейные напряжения сдвинуты друг относительно друга по фазе. Этот фазовый сдвиг постоянен и равен 120 градусам.

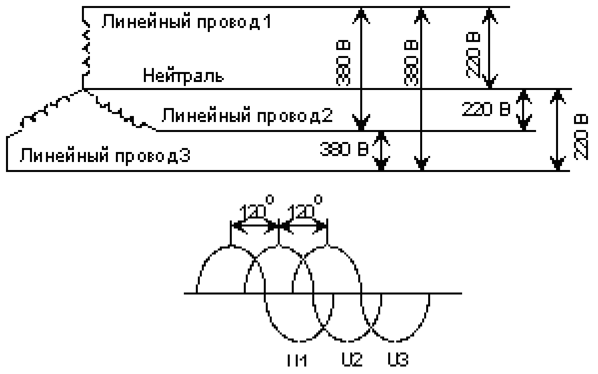


Рис 1.1.Трехфазная система переменного тока.

После трансформаторной подстанции напряжение через распределительные щитки или (на предприятиях) распределительные пункты поступает к потребителям.

Некоторые потребители (электродвигатели, промышленное оборудование, большие ЭВМ и мощное коммуникационное оборудование) рассчитаны на непосредственное подключение к трехфазной электрической сети. К ним подводятся четыре провода (не считая защитного заземления).

Маломощные потребители (персональные компьютеры, бытовые приборы, офисная техника и т.д.) рассчитаны на однофазную электрическую сеть. К ним подводят два провода (не считая защитного заземления). В подавляющем числе случаев один из этих проводов - линейный, а другой - нейтральный. Напряжение между ними по стандарту равно 220 В.

Приведенные выше действующие значения напряжения не исчерпывают полностью параметры электрической сети.

К сети постоянно подключаются новые потребители электроэнергии (ток или нагрузка в сети увеличивается) или отключаются какие-либо потребители (в результате ток или нагрузка сети уменьшается). При увеличении нагрузки напряжение в сети падает, а при уменьшении нагрузки напряжение в сети возрастает.

Для уменьшения влияния изменения нагрузки на напряжение, на понижающих подстанциях существует автоматическая система регулирования напряжения. Она предназначена для поддержания постоянного (в определенных пределах и с определенной точностью) напряжения при изменении нагрузки в сети. Регулирование осуществляется за счет перекоммутации обмоток мощных понижающих трансформаторов.

Частота переменного тока задается частотой вращения генераторов на электростанциях. При увеличении нагрузки частота стремится слегка уменьшиться, система регулирования электростанции увеличивает расход рабочего тела через турбину, и частота вращения генератора восстанавливается.

Разумеется ни одна система регулирования (напряжения или частоты) не может работать идеально, и в любом случае пользователю электрической сети нужно смириться с некоторыми отклонениями характеристик сети от номинальных значений.

Наиболее важные показатели качества электроэнергии - это отклонение напряжения от номинального значения, коэффициент несинусоидальности напряжения, отклонение частоты от 50 Гц.

1.2. Основные понятия и анализ задачи электроэнергетики.

Показатель, по величине которого оценивают, является ли решение оптимальным, называется *критерием оптимальности.* В качестве критерия оптимальности наиболее часто принимается *экономический критерий,* представляющий собой минимум затрат (финансовых, сырьевых, энергетических, трудовых) на реализацию поставленной задачи. При заданной или ограниченной величине указанных затрат экономический критерий выражается в получении максимальной прибыли.

В электроэнергетике в зависимости от требований поставленной задачи могут приниматься и другие критерии оптимальности, в частности:

критерий надежности электроснабжения;

критерий качества электроэнергии;

критерий наименьшего отрицательного воздействия на окружающую среду (экологический критерий).

Решение оптимизационной задачи включает в себя следующие этапы:

Сбор исходной информации (исходных данных).

Составление математической модели, под которой понимается формализованное математическое описание решаемой задачи.

Выбор метода решения, определяемого видом математической модели.

Выполнение математических вычислений, поручаемое, как правило, компьютеру.

Анализ решения задачи.

Рассмотрим подробнее эти этапы поиска оптимального решения. Очевидно, что без достоверной исходной информации остальные этапы поиска оптимального решения не имеют смысла. Без достоверной информации никакие, даже самые точные вычислительные методы и сверхмощные компьютеры не дадут достоверного оптимального решения. Как говорится, «Что посеешь, то и пожнешь». При сборе исходной информации необходимо правильно разделить информацию на главную и второстепенную, а также оценить категорию принимаемой исходной информации. Исходная информация может быть определенной и однозначной. Такая информация и называется *определенной* или *детерминированной.*

Исходная информация может носить случайный характер и подчиняться законам теории вероятностей. Такая информация и называется *случайной*.

Исходная информация может носить неопределенный характер и не подчиняться законам теории вероятностей. Такая информация называется *неопределенной* или *недетерминированной.*

Рассмотрим существующее промышленное предприятие. Мощность, потребляемая предприятием, может быть непосредственно измерена ваттметром. Такая информация будет детерминированной.

Если предприятие проектируется, то мощность, которую будет потреблять предприятие, непосредственно измерить невозможно. О величине этой мощности можно судить лишь с некоторой вероятностью, имея, например, статистические данные об аналогичных объектах. Такая информация будет случайной

Если аналогичные объекты отсутствуют, о величине мощности, которую будет потреблять предприятие, нельзя судить ни однозначно, ни с какой-то вероятностью. В этом случае информация будет недетерминированной.

Формализованное математическое описание оптимизационной задачи, другими словами, математическая модель включает в себя

целевую функцию; ограничения; граничные условия

Целевая функция представляет собой математическую запись критерия оптимальности. При решении оптимизационной задачи ищется экстремум целевой функции, например минимальные затраты или максимальная прибыль. Обобщенная запись целевой функции имеет следующий вид.

$$Z(x_1, x_2, ..., x_n) \to extr \qquad (1.1)$$

где $(x_1, x_2, ..., x_n)$ – искомые переменные, значения которых вычисляются в процессе решения задачи; общее количество переменных равно n.

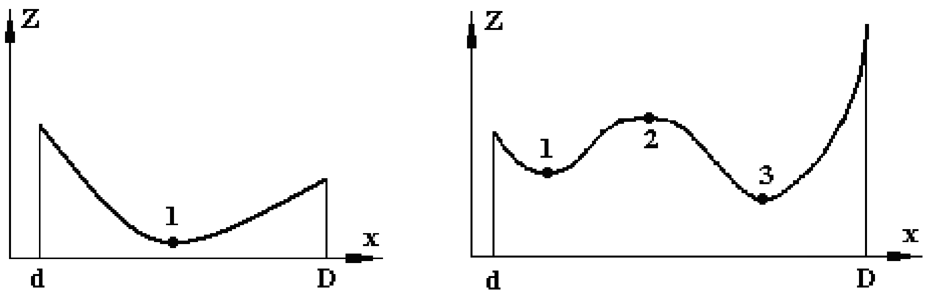
Искомые переменные по своему характеру делятся на непрерывные, дискретные и целочисленные. Если переменная может принимать любые значения, такая переменная называется *непрерывной*. Примером непрерывной переменной может служить мощность, передаваемая по линии электропередачи.

Если переменная может принимать только значения целых чисел, такая переменная называется *целочисленной*. Примером целочисленной переменной может служить количество трансформаторов для промышленным предприятием.

Если переменная может принимать только определенные значения, такая переменная называется *дискретной*. Примером дискретной переменной может служить искомая мощность трансформатора или искомое сечение линии электропередачи. Значения таких величин регламентируются ГОСТ ами. Например, мощности трансформаторов составляют ряд … 630, 1000, 1600, … кВ·А, а сечения линии электропередачи – ряд … 50, 70, 95 … мм^2. Распространенной задачей с дискретными переменными является задача выбора варианта из числа заданных.

Зависимость между переменными в целевой функции (1.1) может быть *линейной* или *нелинейной*. Напомним, что линейной называется такая зависимость, в которую переменные x_i (i=1, 2, 3, … n) входят только в первой степени и с этими переменными выполняются только действия сложения, вычитания и умножения на постоянный коэффициент. Во всех других случаях зависимость будет нелинейной.

Нелинейная целевая функция в заданном диапазоне изменения переменных может иметь один экстремум или несколько экстремумов. В первом случае функция будет *одноэкстремальной*, во втором – *многоэкстремальной*. На рис. 1.2 приведены примеры одноэкстремальной (один минимум) и многоэкстремальной (два минимума и один максимум) функции $Z(x)$ одной переменной в диапазоне изменения этой переменной $d \le x \le D$.



а) б)

Рис. 1.2. Одноэкстремальная (а), многоэкстремальная (б)

В случае многоэкстремальной функции каждый экстремум называется *локальным*. У многоэкстремальных функции ищется глобальный экстремум(наименьший минимум или наибольший максимум). Так при отыскании минимума функции, приведенной на рис. 1.2,б ищется глобальный минимум, отвечающий точке 3.

Ограничения представляют собой различные технические, экономические, экологические условия, учитываемые при решении задачи. Ограничения представляют собой зависимости между переменными $x_1,\ x_2,\ \ldots x_n$, задаваемые в форме неравенств или равенств.

$$\begin{cases} f_1\left(x_1,\ x_2,\ \ldots x_n\right) \le \mathbf{b}_1, \\ f_2\left(x_1,\ x_2,\ \ldots x_n\right) \le \mathbf{b}_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ f_m\left(x_1,\ x_2,\ \ldots x_n\right) \le \mathbf{b}_m \end{cases} \qquad (1.2)$$

Общее количество ограничений равно m. Правые части ограничений, представляющие собой постоянные коэффициенты bj (j=1, 2, … m), называются свободными членами.

Как и в выражении целевой функции (1.1), зависимости между переменными в системе ограничений (1.2) могут быть линейными и нелинейными.

Наличие в системе ограничений (1.2) соотношений в форме неравенств (неполных равенств) создает дополнительные трудности при решении оптимизационной задачи, поскольку в отличие от строгого равенства неравенства представляют собой в некотором роде неопределенность. Например, в неравенстве

$$2x_1 + 3x_2x_3 - x_4 < 4$$

нет определенности на сколько его левая часть меньше 4.

Понятно стремление перейти от ограничений неравенств к равенствам. Для такого перехода используется следующий искусственный прием. Пусть имеем указанное выше неравенство, левая часть которого на неизвестную заранее величину меньше 4. Обозначим эту неизвестную величину как дополнительную неотрицательную переменную $x5$ и добавим ее к левой части неравенства. Последнее обращается в строгое равенство

$$2x_1 + 3x_2x_3 - x_4 + x_5 = 4$$

Совершенно аналогично неполное равенство типа

$$2x_1 + 3x_2x_3 - x_4 \leq 4$$

обращается в строгое равенство

$$2x_1 + 3x_2x_3 - x_4 + x_5 = 4$$

Соотношения типа

$$2x_1 + 3x_2x_3 - x_4 > 4 \quad \text{и} \quad 2x_1 + 3x_2x_3 - x_4 \geq 4$$

после изменения знаков правой и левой частей сводятся к уже рассмотренным случаям.

Таким образом, за счет введения дополнительных переменных все неравенства в системе ограничений (1.2) заменяются строгими равенствами. При этом общее количество n искомых переменных увеличивается.

Предположим, что все m ограничений являются равенствами. При $n = m$ система (1.2) имеет единственное решение. Например, одно уравнение m=1 с одним неизвестным n=1

$$2x_1 = 4$$

имеет единственное решение x_1 =2. Поэтому в случае $n = m$ нет места оптимизации.

При $n < m$ система (1.2) не имеет решения и, следовательно, выбирать оптимальное решение не из чего. Например, система из двух уравнений m=2 с одним неизвестным n=1.

В качестве примера задачи выпуклого программирования рассмотрим простейшую среди задач об оптимальном ведении режима работы энергосистемы.

Рассматривается изолированная энергосистема, состоящая из теп-теплоэлектростанций, связанных линиями передач с узлом, в котором сосредоточена нагрузка. Ставится задача распределения активных мощностей между электростанциями в заданный момент времени. Распределение осуществляется по критерию минимизации суммарных топливных затрат на генерацию активной мощности.

Обозначим через x_i активную мощность, генерируемую на $i-$ й станции. Мощности x_i заключены в пределах α_i и β_i, определяемых техническими условиями: $\alpha_i \le x_i \le \beta_i$. Кроме того, должно соблюдаться условие баланса мощностей, т. е. генерируемая общая мощность должна соответствовать потребляемой мощности Р с учетом общих потерь π в линиях передач:

$$\sum_{i=1}^{m} x_i = P + \pi$$

Топливные затраты на генерацию мощности x_i представляют собой функцию $T_i(x_i)$, выпуклую на отрезке $[\alpha_i, \beta_i]$.

Таким образом, задача принимает вид:

$$\min \sum_{i=1}^{m} T_i\left(x_i\right)$$

при условиях

$$\sum_{i=1}^{m} x_i = P + \pi,\ \alpha_i \le x_i \le \beta_i$$

Построенная модель является типичной задачей выпуклого программирования с линейными ограничениями. Решение этой задачи дает весьма грубое приближение к действительно оптимальному режиму работы энергосистемы. В реальной ситуации нельзя считать всю нагрузку сосредоточенной в одном узле, а следует рассматривать п узлов. Кроме того, потери в системе, естественно, не являются константой, а зависят от величин передаваемых мощностей и параметров линий передач.

ГЛАВА 2. МЕТОДЫ РЕШЕНИЯ МНОГОКРИТЕРИАЛЬНЫЕ ОПТИМИЗАЦИОННЫЕ ЗАДАЧИ ЭЛЕКТРОЭНЕРГЕТИКИ

2.1. Многокритериальные оптимизационные задачи

Рассмотренные выше решения оптимизационных задач выполнялись по одному критерию (по одной целевой функции). На практике не всегда удается свести задачу к одному критерию, поскольку желаемых целей может быть несколько.

Задачи, в которых оптимизация проводится по нескольким критериям, называют задачами *многокритериальной оптимизации.* Такая оптимизация представляет собой попытку найти компромисс между принятыми критериями.

Важным моментом нахождения такого компромисса является назначение *коэффициентов веса* каждого критерия. В конечном итоге решение многокритериальной задачи сводится к оптимизации по одному обобщенному критерию, в который входят все принятые критерии со своими весовыми коэффициентами.

Существует достаточно много способов определения весовых коэффициентов. Рассмотрим один из них, а именно, способ экспертных оценок. Суть этого способа заключается в следующем.

Пусть для решения оптимизационной задачи приняты, например, три критерия (критерий *A*, критерий *B* и критерий *C*). Собирается группа экспертов – специалистов в той области, к которой относится оптимизационная задача. Пусть группа экспертов состоит, например, из трех человек (1-й эксперт, 2-й эксперт и 3-й эксперт). Каждому эксперту предлагается оценить в баллах от 0 до 1 каждый критерий. При этом выдвигается условие, чтобы сумма баллов каждого эксперта по всем критериям была бы равна 1

В табл. 2.1.1 представлены результаты экспертизы. В качестве весового коэффициента *i*-го критерия (*i*=*A*, *B*, *C*) принимается среднее значение оценок каждого эксперта по этому критерию (последняя строка табл. 2.1.1).

Т а б л и ц а 2 . 1 . 1

	К р и т е р и и			
Эксперт	*A*	*B*	*C*	Сумма
1-й	0,2	0,2	0,6	1,0
2-й	0,4	0,3	0,3	1,0
3-й	0,3	0,2	0,5	1,0
Коэф.вес	0,3	0,23	0,47	1,0

Оптимизация по обобщенной целевой функции

Одним из возможных решений многопараметрической задачи является оптимизация по обобщенной целевой функции, в которую входят все принятые к рассмотрению критерии со своими весовыми коэффициентами. Эта обобщенная функция записывается следующимобразом

$$Z_{об} = \sum_{k=1}^{s} \frac{a_k Z_k}{Z_{кнорм}} \rightarrow \max$$

$Z_k - k$ - целевая функция, выраж   ающая *k*-й критерий;

$Z_{кнорм}$ - нормированное значение *k*-й целевой функции;

α_k - коэффициент веса *k*-й целевой функции;

s - количество целевых функций (принятых критериев).

Если *k*-я целевая функция максимизируется, перед ней под знаком суммы ставится плюс. Если *k*-я целевая функция минимизируется, перед ней под знаком суммы ставитсяминус

Весовые коэффициенты могут быть определены, например, с помощью экспертных оценок. Нормированное значение *k*-й целевой функции $Z_{кнрмм}$ принимается по результатам решения оптимизационной задачи только по одному *k*–му критерию

Целевые функции в общем случае имеют разные единицы измерения. Поэтому в (2.1.1) введено деление каждой целевой функции на ее нормированное значение. Такое действие приводит все целевые функций к единой размерности (к относительным единицам, о.е.)

Составление ограничений и граничных условий для многокритериальной задачи не имеет специфических особенностей по сравнению с однокритериальной задачей.

В общем виде математическая формулировка многокритериальной задачи выглядит следующим образом.

Требуется найти значения действительных переменных $x_1,\ldots, x_n$, при которых *целевые функции*

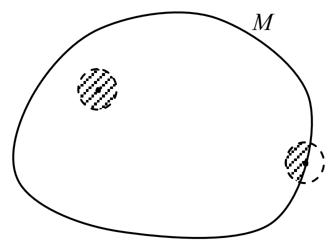
$$L_1(X),\ldots,\quad L_p(X)$$

принимают экстремальные значения при ограничениях:

$$g_i(X) \le 0, i = 1,\ldots,m,$$

Множество Парето

Напомним некоторые определения. Пусть на плоскости (или в пространстве) дано некоторое множество точек M. Точка P называется *внутренней точкой множества* M, если существует такая окрестность этой точки, которая целиком состоит из точек данного множества. Если же в любой окрестности точки P имеются точки, как принадлежащие, так и не принадлежащие множеству M, то точка P называется *граничной точкой множества* M. Совокупность всех граничных точек данного множества M называется его *границей*.



Если множество M не содержит ни одной своей граничной точки, то оно называется *открытым* (то есть любая точка открытого множества является внутренней). Если множество M содержит все свои граничные точки, то оно называется *замкнутым*. В дальнейшем будут рассматриваться только замкнутые множества.

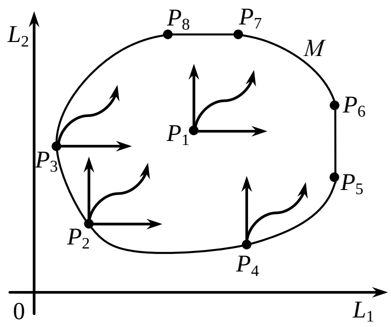


Рис. 2.1.1. Возможные перемещения

Рассмотрим на плоскости OL_1L_2 множество M. Пусть P — произвольная точка этого множества. Возможно ли во множестве M перемещение точки P в близкую ей точку так, чтобы при этом увеличились обе ее координаты? Если P — внутренняя точка, то такое перемещение возможно. Если P — граничная точка, то такое перемещение не всегда возможно. Иллюстрацией служит Рис. 2.1.1. Требуемое перемещение точек P_1, P_2, P_3, P_4 возможно, а ни одна из точек как отрезков P_5P_6 и P_7P_8, так и дуги P_6P_7 такому перемещению подвергнута быть не может. Действительно, при перемещении любой точки

- вертикального отрезка P_5P_6 может увеличиваться лишь координата L_2 этой точки (координата L_1 при этом останется неизменной);
- горизонтального отрезка P_7P_8 может увеличиваться лишь координата L_1 (координата L_2 при этом останется неизменной);
- дуги P_6P_7 увеличение одной координаты влечет уменьшение другой.

Таким образом, каждая точка множества M попадает в один из трех следующих классов.

- Первый класс содержит точки, каждую из которых можно переместить так, чтобы при этом увеличились обе ее координаты, а сама точка осталась во множестве M (в этот класс попадают все внутренние точки множества M и некоторые его граничные точки (например, P_2)).
- Второй класс содержит точки, каждую из которых можно переместить во множестве M лишь при условии увеличения только одной из ее координат при сохранении значения второй (точки вертикального отрезка P_5P_6 и точки горизонтального отрезка P_7P_8).

- Третий класс содержит точки, каждую из которых можно переместить во множестве *M* лишь при условии уменьшения хотя бы одной из координат (точки дуги P_6P_7).

Множество точек третьего класса называют *границей* (*множеством*) *Парето* данного множества *M*. Часто говорят, что граница Парето множества *M* — это множество точек, из которых нельзя переместиться на «север», «восток» или «северо-восток», оставаясь во множестве *M*. Свойства множества Парето изучены достаточно подробно (см., например, [10]), разработаны методы и алгоритмы его построения. Считается, что наилучшие решения многокритериальной задачи следует искать именно среди множества Парето. Поэтому построение множества Парето нередко считают первым необходимым шагом в решении любой многокритериальной задачи. **Задача линейной многокритериальной максимизации с двумя переменными и двумя целевыми функциями.** Указанная задача является частным случаем многокритериальной задачи в случае $p = 2$. Сформулируем ее. Пусть на плоскости Ox_1x_2 задано множество $\overline{X}$ (Рис.2.1.2) и в каждой точке этого множества определены две непрерывные функции $L_1 = f_1(x_1, x_2)$ и $L_2 = f_2(x_1, x_2)$. Необходимо найти значения переменных, при которых указанные функции принимают наибольшие значения. Формулировку задачи максимизации с двумя целевыми функциями можно записать более компактно:

$$f_1(x_1, x_2) \to \max;$$

$$f_2(x_1, x_2) \to \max$$

при ограничениях:

$$(x_1, x_2) \in \overline{X}.$$

Попытаемся ее решить. Изобразим на плоскости OL_1L_2 все точки, координаты которых удовлетворяют условиям $L_1 = f_1(x_1, x_2)$, $L_2 = f_2(x_1, x_2)$ и $(x_1, x_2) \in \overline{X}$. Полученное множество обозначим через $\overline{L}$ (Рис. 2.1.32.1.3).

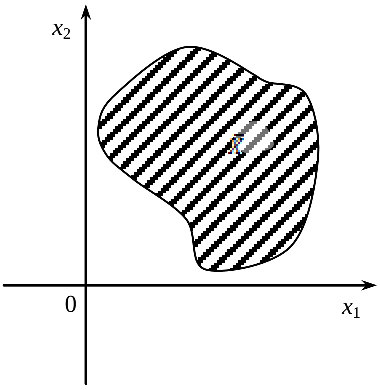


Рис.2.1.2. ОДР на плоскости Ox_1x_2

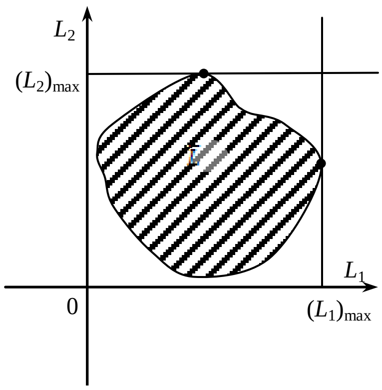


Рис. 1. ОДР на плоскости OL_1L_2

Из Рис. 2.1.32.1.3 видно, что $(L_1)_{\max}$ — наибольшее значение L_1 — и $(L_2)_{\max}$ — наибольшее значение L_2 — достигаются в разных точках. При этом $((L_1)_{\max},\ (L_2)_{\max}) \notin \overline{L}$. Это означает, что задача неразрешима — не существует оптимального решения, которое одновременно максимизировало бы обе целевые функции. Поэтому нужно искать Парето-оптимальное решение. Как уже выше отмечалось, наилучшие решения многокритериальной задачи следует искать среди множества Парето. Рассмотрим два метода нахождения недоминируемого решения, связанных с множеством Парето:

1. Метод (последовательных) уступок.
2. Метод идеальной точки.

В рассматриваемом случае множество Парето составлено из допустимых точек задачи, которые не могут быть перемещены в пределах допустимого множества с улучшением сразу по двум критериям: улучшение значения одного критерия влечет ухудшение значения другого.

Метод (последовательных) уступок заключается в том, что ЛПР, работая в режиме диалога со специалистом, анализирует точки на границе Парето и выбирает одну из них — компромиссную.

Метод идеальной точки заключается в нахождении на границе Парето точки, ближайшей к *точке утопии*, задаваемой ЛПР. Как правило, ЛПР формулирует цель в виде определенных показателей, и часто в качестве координат целевой точки выбирается комбинация наилучших значений всех критериев (в данном случае — точка с координатами $((L_1)_{\max}, (L_2)_{\max})$). Обычно эта точка не реализуется при заданных ограничениях, поэтому ее и называют точкой утопии.

Замечание 1. Задачу максимизации можно путем умножения целевой функции на (–1) преобразовать в задачу минимизации, решаемую при тех же самых ограничениях. Это связано с наличием следующего свойства: функция $(-f)$ достигает наибольшего значения в тех точках, в которых функция f принимает наименьшее значение, и наоборот. Это означает, что условия $[f \rightarrow \min]$ и $[(-f) \rightarrow \max]$ равносильны. Следовательно, поменяв знак целевой функции на противоположный, любую двухкритериальную задачу можно свести к задаче максимизации с двумя целевыми функциями.

2.2. Решение задач линейного программирования симплекс–методом

Графический метод решения задачи линейного программирования применяется только в случае двух или трех переменных. Симплексный метод (или симплекс-метод) в отличие от него является универсальным, так как позволяет решить общую задачу линейного программирования (ОЗЛП) практически с любым количеством переменных. Но для этого она должна быть приведена к каноническому виду или к канонической задаче линейного программирования (КЗЛП). Множество канонических задач является подмножеством всех задач линейного программирования. Перечислим отличительные черты КЗЛП:

Все ограничения в системе ограничений являются равенствами. Требование неотрицательности наложено на все переменные, входящие в систему ограничений.

Целевая функция минимизируется. Надо отметить, что это требование влечет только лишь признак прекращения вычислений, который мы укажем при изложении алгоритма симплекс-метода, и не является принципиальным.

Покажем, что любая задача линейного программирования может быть приведена к каноническому виду. Для этого необходимо выполнить следующие преобразования ОЗЛП:

Всякое условие вида $\sum_{j=1}^{n} a_{sj}x_j \le b_s$, где s = 1 или 2, или …, или n заменяется условием $\sum_{j=1}^{n} a_{sj}x_j + y_s = b_s, y_s \ge 0.$

Всякое условие вида $\sum_{j=1}^{n} a_{sj}x_j \ge b_s$, где s = 1 или 2, или …, или n заменяется условием $\sum_{j=1}^{n} a_{sj}x_j - y_s = b_s, y_s \ge 0.$

Все переменные xp, на которые не наложены требования неотрицательности, подставляются в равенства, полученные на двух предыдущих шагах, и в целевую функцию в виде xp = wp – vp, wp ≥ 0, vp ≥ 0.

После преобразований 1—3 все новые переменные удобно обозначить буквой x, пронумеровав их по порядку введения, начиная с номера m + 1. Причем на шаге 3 переменную wp удобно обозначить xp, а vp — буквой x с первым свободным номером. Возможно, за счет введения новых переменных на шаге 3 изменится и целевая функция. Новую целевую функцию обозначим L1(X).

В случае оптимизации вида L1(X)→max воспользуемся равенством max L1(X) = –min(–L1(X)). Поэтому введем новую целевую функцию $\overline{L}$(X) = –L1(X), решим задачу при условии $\overline{L}$(X)→min, после чего сделаем замену max L1(X) = –min(–$\overline{L}$(X)).

Пример 1. Общая задача линейного программирования

$$\begin{cases} x_1 + x_2 + x_3 \le 7, \\ 2x_1 - x_2 + x_3 \ge 3, \\ x_1 - x_2 + 9x_3 = 16, \\ x_1 \ge 0, \end{cases}$$

$$L(X) = x_1 + x_2 - x_3 \to \max$$

приводится к каноническому виду согласно шагам 1—4.

$$x_1 + x_2 + x_3 + y_1 = 7, y_1 \ge 0.$$

$$2x_1 - x_2 + x_3 - y_2 = 3, y_2 \ge 0.$$

x2 = w2 – v2, w2 ≥ 0, v2 ≥ 0; x3 = w3 – v3, w3 ≥ 0, v3 ≥ 0.

Теперь обозначим y1 как x4, y2 — x5, w2 — x2, v2 — x6, w3 — x3, v3 — x7. Задача оптимизации при этом примет вид:

$$L_1(X) = x_1 + (x_2 - x_6) - (x_3 - x_7) \to \max.$$

$$\overline{L}(X) = -x_1 - x_2 + x_3 + x_6 - x_7 \to \min.$$

Пример 2. Из общей задачи линейного программирования

$$\begin{cases} x_1 + x_2 + x_3 - x_4 - x_5 \le 7, \\ 2x_1 - x_2 + x_3 + x_4 - x_5 \ge 3, \\ x_1 - x_2 + 9x_3 + x_4 - 8x_5 = 16, \\ x_j \ge 0, j = 1,...,5, \end{cases}$$

$$L(X) = -x_1 - x_2 + x_3 + x_4 - x_5 \to \min$$

путем введения новых неотрицательных переменных получаем каноническую задачу:

$$\begin{cases} x_1 + x_2 + x_3 - x_4 - x_5 + x_6 = 7, \\ 2x_1 - x_2 + x_3 + x_4 - x_5 - x_7 = 3, \\ x_1 - x_2 + 9x_3 + x_4 - 8x_5 = 16, \\ x_j \ge 0, j = 1,...,7, \end{cases}$$

$$L(X) = -x_1 - x_2 + x_3 + x_4 - x_5 \to \min.$$

Ясно, что новые переменные можно сразу обозначать буквой x.

Легко показать, что в результате преобразований 1—4 из общей задачи линейного программирования получается каноническая задача, решение которой тесно связано с решением исходной задачи. Действительно, пусть количество переменных при переходе к КЗЛП с n увеличилось до n + t и пусть Z = $\left(x_1, x_2,..., x_n, x_{n+1},..., x_{n+t}\right)$ — какое-нибудь решение КЗЛП. Ясно, что после отбрасывания из Z последних t компонент, согласно правилу их введения, получается решение исходной задачи линейного программирования. Поскольку в целевую функцию отброшенные компоненты не входят вообще или разность двух из них в точности равна значению некоторой исходной переменной, то значения $\overline{L}\left(x_1, x_2,..., x_n, x_{n+1},..., x_{n+t}\right)$ и $L\left(x_1, x_2,..., x_n\right)$ равны по модулю. И наоборот, располагая каким-нибудь решением $\left(x_1, x_2,..., x_n\right)$ исходной задачи, можно вычислить и соответствующее решение КЗЛП $\left(x_1, x_2,..., x_n, x_{n+1},..., x_{n+t}\right)$, посчитав, согласно правилам ввода, значения недостающих t компонент. Так как выполняется равенство $\left|\overline{L}\right| = \left|L\right|$, то оптимальное решение Z* КЗЛП Z*= ($x_1^*, x_2^*,..., x_n^*, x_{n+1}^*,..., x_{n+t}^*$) после отбрасывания последних *t* компонент даст оптимальное решение исходной задачи: X^*= ($x_1^*, x_2^*,..., x_n^*$).

Таким образом, преобразования (1)—(4) приводят к КЗЛП, которую в общем виде можно записать:

$$\begin{cases} \sum_{j=1}^{n+t} a_{ij} x_j = b_i, \ i = 1,..., m; \\ x_j \geq 0, \ j = 1,..., n+t; \end{cases} \qquad \begin{matrix} (1) \\ (2) \\ (3) \end{matrix}$$

$$\overline{L}(X) = c_0 + \sum_{j=1}^{n+t} c_j x_j \to \min. \qquad (4)$$

Пусть ранг системы линейных уравнений (1) в этой задаче равен *r*. Это означает, что *r* переменных этой системы, называемых базисными, могут быть выражены через остальные *n* + *t* – *r* переменных, называемых свободными. Напомним, что решение, в котором все свободные переменные равны нулю, а базисные принимают соответствующие неотрицательные значения, называется опорным планом.

В теории линейного программирования доказано, что оптимальное значение целевая функция (3) принимает на множестве опорных планов (решений) КЗЛП, которое конечно. Таким образом, перебрав все опорные решения, можно указать то из них, на котором целевая функция оптимальна. Но даже при небольшом числе переменных, количество опорных планов может быть очень велико, и полный их перебор займет много времени. Идея же симплексного метода, или метода последовательного улучшения плана, заключается в том, что начиная с некоторого исходного опорного решения (плана) осуществляется последовательное целенаправленное перемещение по опорным планам в сторону оптимального значения целевой функции. Не излагая теории симплексного метода, укажем его алгоритм. Сначала рассмотрим алгоритм, применяющийся для расчетов вручную.

Жордановы исключения. Поскольку переход от одного опорного плана к другому означает перевод некоторой базисной переменной в свободные, а соответствующей свободной переменной в базисные, рассмотрим отдельно эту процедуру в общем виде и запишем ее алгоритм. Для понимания сути преобразований возьмем систему двух линейных алгебраических уравнений с пятью переменными, имеющую ранг 2. Это означает, что две базисные переменные могут быть выражены через три свободные. Для того, чтобы удобно было следить за перемещением базисной переменной в свободную и наоборот, обозначим свободные переменные x_1, x_2, x_3, а базисные — y_1, y_2:

$$\begin{cases} y_1 = a_{11}x_1 + a_{12}x_2 + a_{13}x_3, \\ y_2 = a_{21}x_1 + a_{22}x_2 + a_{23}x_3. \end{cases}$$

Переведем базисную переменную y_1 в свободные, а свободную переменную x_2 — в базисные, то есть выразим x_2, y_2 через x_1, y_1, x_3. При этом новую базисную переменную будем писать на месте старой, то есть в первой строке системы, и новую свободную — на месте старой, то есть во втором столбце. Пусть коэффициент a_{12} при свободной переменной x_2 отличен от нуля.

1. Все независимые (свободные) переменные запишем со знаком «–»:

$$\begin{cases} y_1 = (-a_{11})(-x_1) + (-a_{12})(-x_2) + (-a_{13})(-x_3), \\ y_2 = (-a_{21})(-x_1) + (-a_{22})(-x_2) + (-a_{23})(-x_3). \end{cases}$$

2. Для удобства обозначим $-a_{ij} = \alpha_{ij}$:

$$\begin{cases} y_1 = \alpha_{11}(-x_1) + \alpha_{12}(-x_2) + \alpha_{13}(-x_3), \\ y_2 = \alpha_{21}(-x_1) + \alpha_{22}(-x_2) + \alpha_{23}(-x_3). \end{cases}$$

3. Поделим первую строчку на α_{12}:

$$\frac{y_1}{\alpha_{12}} = \frac{\alpha_{11}}{\alpha_{12}}(-x_1) + 1 \cdot (-x_2) + \frac{\alpha_{13}}{\alpha_{12}}(-x_3).$$

4. Выразим из этого уравнения x_2:

$$x_2 = \frac{\alpha_{11}}{\alpha_{12}}(-x_1) + \frac{1}{\alpha_{12}} \cdot (-y_1) + \frac{\alpha_{13}}{\alpha_{12}}(-x_3).$$

5. Сделаем преобразования во втором уравнении, заменив в нем x_2 на указанное выше выражение:

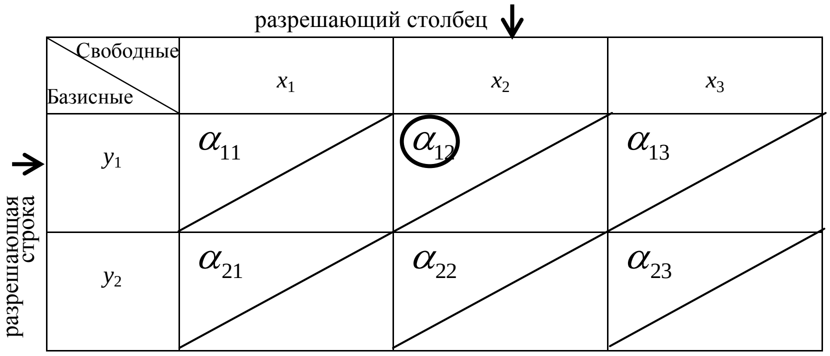
$$y_2 = \left(\alpha_{21} - \alpha_{22} \cdot \frac{\alpha_{11}}{\alpha_{12}}\right)(-x_1) + \left(-\frac{\alpha_{22}}{\alpha_{12}}\right) \cdot (-y_1) + \left(\alpha_{23} - \alpha_{22} \cdot \frac{\alpha_{13}}{\alpha_{12}}\right)(-x_3).$$

В результате проделанных шагов 1—5 мы выразили x_2, y_2 через x_1, x_3, y_1. Причем новые свободные переменные записаны со знаком «–», то есть полученная система

$$\begin{cases} x_2 = \dfrac{\alpha_{11}}{\alpha_{12}}(-x_1) + \dfrac{1}{\alpha_{12}} \cdot (-y_1) + \dfrac{\alpha_{13}}{\alpha_{12}}(-x_3), \\ y_2 = \left(\alpha_{21} - \alpha_{22} \cdot \dfrac{\alpha_{11}}{\alpha_{12}}\right)(-x_1) + \left(-\dfrac{\alpha_{22}}{\alpha_{12}}\right) \cdot (-y_1) + \left(\alpha_{23} - \alpha_{22} \cdot \dfrac{\alpha_{13}}{\alpha_{12}}\right)(-x_3) \end{cases}$$

снова готова к описанным преобразованиям типа 3—5. Проделанная процедура называется *жордановыми преобразованиями*, или исключениями.

Жордановы исключения очень удобно проводить с помощью специальных таблиц. Количество строк в таблице равно количеству базисных переменных (т. е. рангу системы), а количество столбцов — числу свободных переменных. Если справа от знака равенства имеются свободные члены, то



появляется еще один столбец и для них. Преобразуются они, очевидно, по тем же правилам, что и остальные коэффициенты. Все клетки исходной таблицы, содержащие числовые величины, делятся по диагонали (таблица 5).

В верхние треугольники исходной таблицы помещаются коэффициенты системы, полученной после шага 2. Столбец, содержащий коэффициенты при свободной переменной, которую необходимо перевести в базисные, будем называть разрешающим столбцом, а строку с соответствующей базисной переменной — разрешающей строкой и отмечать стрелками. На пересечении разрешающих строки и столбца находится разрешающий элемент. В таблице он обведен кружком.

В алгоритме жордановых исключений запишем последовательность преобразований коэффициентов системы на шагах 3—5.

Преобразование исходной таблицы (таблица 5):

1. Разрешающий элемент заменяется обратной величиной и записывается внизу.

2. Остальные элементы разрешающей строки делятся на разрешающий элемент и записываются внизу.

3. Остальные элементы разрешающего столбца делятся на разрешающий элемент, меняют знак и записываются внизу.

4. Обозначим буквой *в* — верхний элемент, буквой *н* — нижний элемент, буквой k — номер разрешающей строки, буквой s — номер разрешающего столбца. В остальных, не заполненных внизу, клетках таблицы с номерами i,j рассчитывается величина $н_{ij} = в_{kj} \cdot н_{is}$. Например, $н_{21} = в_{11} \cdot н_{22} = \alpha_{11} \cdot \left(-\frac{\alpha_{22}}{\alpha_{12}} \right)$.

Ниже приведена таблица 6 — исходная таблица, для которой выполнены преобразования (1)—(4):

Таблица 6

разрешающий столбец ↓ (x_2)

Свободные / Базисные	x_1	x_2	x_3
→ y_1	α_{11} / $\frac{\alpha_{11}}{\alpha_{12}}$	α_{12} / $\frac{1}{\alpha_{12}}$	α_{13} / $\frac{\alpha_{13}}{\alpha_{12}}$
y_2	α_{21} / $-\alpha_{11}\frac{\alpha_{22}}{\alpha_{12}}$	α_{22} / $-\frac{\alpha_{22}}{\alpha_{12}}$	α_{23} / $-\alpha_{13}\frac{\alpha_{22}}{\alpha_{12}}$

Переход к новой таблице. Для этого необходимо построить такую же таблицу, что и исходная, поменяв местами рассматриваемые свободную и базисную переменные. Затем выполнить указанные ниже шаги.

1. Нижние элементы разрешающей строки записываются вверху.
2. Остальные элементы разрешающего столбца записываются вверху.
3. В остальных клетках верхние и нижние элементы складываются и

Таблица 7

разрешающий столбец ↓ (y_1)

Свободные / Базисные	x_1	y_1	x_3
→ x_2	$\frac{\alpha_{11}}{\alpha_{12}}$	$\frac{1}{\alpha_{12}}$	$\frac{\alpha_{13}}{\alpha_{12}}$
y_2	$\alpha_{21} - \alpha_{11}\frac{\alpha_{22}}{\alpha_{12}}$	$-\frac{\alpha_{22}}{\alpha_{12}}$	$\alpha_{23} - \alpha_{13}\frac{\alpha_{22}}{\alpha_{12}}$

записываются вверху.

Выше приведена таблица 7 — новая таблица, в которой выполнены преобразования (1)—(3). Она готова к дальнейшим преобразованиям, то есть снова можно менять местами какие-нибудь базисную и свободную переменные при условии, что соответствующий числовой коэффициент при свободной переменной отличен от нуля. К тому же, если вести заполнение исходной таблицы карандашом, стирая величины, в которых уже нет необходимости, то для всех расчетов можно использовать один и тот же табличный шаблон. Это свойство симплекс-таблиц очень удобно при программировании: все расчеты идут в пределах одного двумерного массива, верхние элементы исходной таблицы представляют собой старые значения коэффициентов (элементов массива), а верхние значения преобразованной таблицы — новые значения коэффициентов с теми же номерами. К этому вопросу мы еще вернемся при рассмотрении программы симплекс-метода.

Мы приводим этот алгоритм, не рассматривая подробно теорию симплекс-метода, но там, где это возможно, будем давать обоснование проводимым действиям.

1. Начало. Общая задача линейного программирования приводится к каноническому виду (1)—(3), то есть путем введения новых дополнительных неотрицательных переменных все неравенства, входящие в математическую модель, превращаются в равенства, а все переменные, на которые не наложены требования неотрицательности, представляются в виде разности новых неотрицательных переменных. Целевая функция при этом минимизируется. Алгоритм такого перехода изложен выше.

2. Определяется ранг матрицы A системы уравнений $\sum_{j=1}^{n+t} a_{ij} x_j = b_i, i = 1,\dots,m;$ канонической задачи: пусть $rank\ A = k$. После этого базисные переменные выражаются через свободные. Как правило, базис образуют вновь введенные переменные, но это не обязательно. Допустим, базис образуют первые k переменных. Выразим их через свободные,

записывая, как и при реализации алгоритма жордановых исключений, свободные переменные со знаком «–». Получим систему:

$$\begin{cases} x_1 = b_{10} - b_{1,k+1}x_{k+1} - b_{1,k+2}x_{k+2} - \ldots - b_{1,n+t}x_{n+t}, \\ x_2 = b_{20} - b_{2,k+1}x_{k+1} - b_{2,k+2}x_{k+2} - \ldots - b_{2,n+t}x_{n+t}, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. \\ x_k = b_{k0} - b_{k,k+1}x_{k+1} - b_{k,k+2}x_{k+2} - \ldots - b_{k,n+t}x_{n+t}. \end{cases}$$

Подставим выражения для базисных переменных в целевую функцию. После этого в ней будут только свободные переменные с некоторыми коэффициентами перед ними. В целевой функции также запишем свободные переменные со знаком «–»:

$$\overline{L}(X) = \overline{c_0} + \overline{c_{k+1}}(-x_{k+1}) + \overline{c_{k+2}}(-x_{k+2}) + \ldots + \overline{c_{n+t}}(-x_{n+t}).$$

Для полученной системы и целевой функции составляется таблица для жордановых исключений, т. н. симплекс-таблица (таблица 8):

Таблица 8

Свободные / Базисные	b_{i0}	x_{k+1}	x_{k+2}	…	x_{n+t}
x_1	b_{10}	$b_{1,\,k+1}$	$b_{1,\,k+2}$	…	$b_{1,\,n+t}$
x_2	b_{20}	$b_{2,\,k+1}$	$b_{2,\,k+2}$	…	$b_{2,\,n+t}$
…	…	…	…	…	…
x_k	b_{k0}	$b_{k,\,k+1}$	$b_{k,\,k+2}$	…	$b_{k,\,n+t}$
$\overline{L}(X)$	$\overline{c_0}$	$\overline{c_{k+1}}$	$\overline{c_{k+2}}$		$\overline{c_{n+t}}$

3. Выясняется, имеются ли в последней строке таблицы 8 положительные элементы, кроме $\overline{c_0}$. Пусть, например, коэффициент $\overline{c_{k+1}} > 0.$ Значит, увеличивая переменную x_{k+1}, мы уменьшаем слагаемое $\overline{c_{k+1}}(-x_{k+1})$, а значит, и целевую функцию. Следовательно, опорный план, в котором переменная x_{k+1} имеет нулевое значение, поскольку является свободной, не

оптимален. Но ненулевое значение переменная x_{k+1} может принять, если будет базисной. Значит, ее надо перевести в базисные. Столбец, содержащий положительный коэффициент в последней строке, может стать разрешающим столбцом. Если столбцов с положительными элементами в последней строке несколько, то в качестве разрешающего столбца может быть выбран любой из них. Для уточнения номера разрешающего столбца надо перейти к пункту 4. Если же в последней строке положительных элементов нет, то процесс вычисления завершен, и опорное решение, соответствующее последней таблице, будет оптимальным. Выписываем его, значение целевой функции, даем интерпретацию полученных результатов. Конец.

4. Пусть столбец, который может стать разрешающим, имеет номер j, то есть $\overline{c_j} > 0.$ Это означает, как уже говорилось выше, можно уменьшить значение целевой функции путем увеличения значения x_j, оставляя все другие свободные переменные без изменений, то есть равными нулю. В системе (4), на основе которой построена симплекс-таблица 4, имеем:

$$\begin{cases} x_1 = b_{10} - b_{1j}x_j \geq 0, \\ x_2 = b_{20} - b_{2j}x_j \geq 0, \\ \dots\dots\dots\dots\dots\dots\dots \\ x_k = b_{k0} - b_{kj}x_j \geq 0. \end{cases}$$

Ясно, что увеличивая x_j, надо следить за тем, чтобы x_1, x_2,…, x_k оставались неотрицательными. Здесь возможны два случая:

а) Все коэффициенты $b_{ij} < 0$, $i = 1,\dots, k$. Тогда значение x_j можно увеличивать неограниченно, от этого произведения $-b_{ij}x_j$ будет только расти, и все переменные x_1, x_2,…, x_k будут неограниченно возрастать, то есть требование их неотрицательности не будет нарушено. Целевая функция при этом $\overline{L}(X) = \overline{c_0} + \overline{c_j}(-x_j),\ \overline{c_j} > 0,$ не будет ограничена снизу: $\overline{L}(X) \to -\infty.$

б) Среди b_{ij} , $i = 1,\dots, k$ имеются положительные, и таких чисел может быть несколько. Пусть $b_{rj} > 0$ $r = 1$ или $r = 2$, или …, или $r = k$. Ясно, чтобы выполнялось условие $x_r = b_{r0} - b_{rj}x_j \geq 0$, надо потребовать выполнения равносильного неравенства $x_j \leq \frac{b_{ro}}{b_{rj}}$, то есть переменную x_j можно увеличивать, но только до величины $\frac{b_{ro}}{b_{rj}}$. И такие неравенства должны выполнятся для всех строк j-го столбца, в которых $b_{rj} > 0$. Следовательно, надо для таких строк определить величину

$$K = \min \frac{b_{r0}}{b_{rj}}, \ \ 1 \le r \le k, \ \ \ b_{rj} > 0.$$

Ясно, что x_j можно увеличивать, но не более чем до K. Значит, x_j надо менять местами с той базисной переменной, в которой строке указанный минимум достигается.

Итак, последовательность действий в пункте 4 такова: просматривается столбец, соответствующий $\overline{c_j} > 0,$ выясняется, имеются ли в нем положительные элементы; если ни в одном из таких столбцов положительных элементов нет, то оптимального решения не существует, так как целевая функция не ограничена снизу, конец; если найден хотя бы один столбец, содержащий положительный элемент в последней и какой-нибудь еще строках, то этот столбец — разрешающий. Далее надо перейти к пункту 5.

5. В разрешающем столбце j находится строка с номером i, для которой достигается $\min \frac{b_{r0}}{b_{rj}}, \ \ 1 \le r \le k, \ \ b_{rj} > 0.$ Строка i — разрешающая строка. Перейти к пункту 6.

6. Мен яются местами переменные x_j и x_i. Для этого в последней симплекс-таблице надо выполнить жордановы исключения по соответствующему алгоритму. Вернуться к пункту 3.

2.3 Транспортная задача

Транспортная задача является одной из важных частных задач линейного программирования. Впервые, насколько нам известно, транспортная задача рассматривалась А. Н. Толстым. В американской литературе первая работа, посвященная этой задаче, относится к 1941 г. и принадлежит Хичкоку, в честь которого задача эта иногда именуется в американской печати проблемой Хичкока.

Следует отметить, что в обеих упомянутых работах не было еще законченного метода решения рассматриваемой задачи. Первый общий метод решения транспортной проблемы, получивший в дальнейшем название «метода потенциалов», предложен Л. В. Канторовичем в 1949 г. Метод Канторовича был разработан вне связи с общими методами линейного программирования. Несколько позже тот же метод, исходя из общих соображений, был получен Данцигом.

Специфичность условий задачи транспортировки приводит к тому, что наряду с обычными свойствами задачи линейного программирования она обладает рядом интересных, свойственных лишь ей особенностей.

Постановка задачи. В этом параграфе дается постановка транспортной задачи и обсуждаются некоторые присущие ей особенности. В основном результаты параграфа носят качественный характер. Многие из них будут использованы в последующем при описании методов решения задачи транспортировки. Содержание транспортной задачи состоит в следующем. Пусть в пунктах $A_1, A_2, ..., A_m$ производится некоторый однородный продукт, причем объем производства в пункте A_i составляет a_i единиц. Допустим также, что указанный продукт потребляется в пунктах $B_1, B_2, ..., B_n$, причем объем потребления в пункте B_j составляет b_j единиц продукта. Пункты $A_1, A_2, ..., A_m$ будем называть пунктами производства, пункты $B_1, B_2, ..., B_n$ – пунктами потребления. В задаче предполагается, что суммарный объем производства $\sum_{i=1}^{m} a_i$ совпадает с суммарным объемом потребления $\sum_{j=1}^{n} b_i$. Положим, что из каждого пункта производства возможна транспортировка продукта в любой из пунктов потребления, причем транспортные издержки, приходящиеся на единицу продукта при перевозке его из $i-$ го пункта производства $\left(A_i\right)$ в $j-$ й пункт потребления $\left(B_j\right)$, заданы и составляют c_{ij} денежных единиц. Задача состоит в составлении такого плана перевозок, при котором запросы всех пунктов потребления были бы удовлетворены за счет производства в пунктах $A_1, A_2, ..., A_m$ и при этом суммарные транспортные издержки были бы минимальными. Все условия транспортной задачи удобно записать в виде наглядной табл. 2.1.1

		Пункты потребления						
		B_1	B_2	.....	B_{n-1}	B_n		
Пункты производства	A_1	c_{11}	c_{12}	.....	$c_{1,\,n-1}$	c_{1n}	a_1	Объем производства
	A_2	c_{21}	c_{22}	.....	$c_{2,\,n-1}$	c_{2n}	a_2	
	.	.	.	.	.	.	.	
	.	.	.	.	.	.	.	
	A_m	c_{m1}	c_{m2}	.....	$c_{m,\,n-1}$	c_{mn}	a_m	
		b_1	b_2	.....	b_{n-1}	b_n		
		Объем потребления						

Табл. 2.1.1 заполнена величинами транспортных издержек. В нижней строке помещены объемы потребления пунктов $B_1, B_2, ..., B_n$, в последнем столбце - объемы производства в пунктах $A_1, A_2, ..., A_m$.

Приведем теперь строгую постановку задачи транспортировки. Пусть x_{ij} обозначает количество продукта, перевозимое из A_i в B_j.

Требуется определить совокупность величин $x_{ij}\; i=\overline{1,m},\; j=\overline{1,n}$ (план перевозок), удовлетворяющих условиям:

$$\left.\begin{aligned} \sum_{j=1}^{n} x_{ij} = a_i, i=\overline{1,m}, \\ \sum_{i=1}^{m} x_{ij} = b_j,\; j=\overline{1,n} \end{aligned}\right\} \tag{2.1.1}$$

$$x_{ij} \geq 0 \tag{2.1.2}$$

и такую, что линейная форма

$$L(X)=\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij} \tag{2.1.3}$$

(суммарные транспортные издержки) достигает на ней своего минимума, т. е. для любой другой совокупности $X'=\{x'_{ij}\}$, удовлетворяющей условиям (2.1.1), (2.1.2) имеет место соотношение

$$L(X)\le L(X')$$

Соотношение (2.1.1) представляющий собой балансы мощности в каждом из узлов, являются ограничениями при решении транспортной задачи. Общее количество узлов источников и потребителей $n+m$.

Из теоретический электротехники известно, что для любой электрической сети количество независимых уравнений составленных по 1-му закону Кирхгофа, на единицу меньше количества узлов и составляет $n+m-1$. Следовательно, количество независимых ограничений составляет $n+m-1$. Количество базисных (не равных нулю) переменных равняется количеству независимых ограничений и составляет $n+m-1$. Остальные переменные являются свободными (равными нулю). Количество свободных переменных составляет $nm-(m+n-1)$.

Каждая базисная переменная x_{ij} соответствует присутствию в схеме линии между узлами i и j, поскольку мощность, протекающая между узлами i и j, не равна нулю. Каждая свободная переменная x_{ij} соответствует отсутствию в схеме линии между узлами i и j, поскольку мощность, протекающая между узлами i и j, равна нулю.

3. ГЛАВА. ЗАДАЧИ ЭЛЕКТРОЭНЕРГЕТИКИ В СРЕДЕ MATLAB.

3.1 Основы работы в среде Matlab.

Современная компьютерная математика предлагает целый набор интегрированных программных систем и пакетов программ для автоматизации математических расчетов: MATLAB, Eureka, Gauss, TK Solver!, Derive, Mathcad, Mathematica, Maple V и др. Среди них, MATLAB *занимает* одно из *лидирующих мест*.

MATLAB — одна из старейших, тщательно проработанных и проверенных временем систем автоматизации математических расчетов, построенная на расширенном представлении и применении матричных операций. Это нашло отражение в названии системы — MATrix LABoratory — матричная лаборатория.

MATLAB является важным инструментом для решения широкого спектра научных и прикладных задач, в таких областях как: моделирование объектов и разработка систем управления, проектирование коммуникационных систем, обработка сигналов и изображений, измерение сигналов и тестирование, финансовое моделирование, вычислительная биология и для многокритериальной оптимизации в том числе.

В настоящее время MATLAB далеко вышла за пределы специализированной матричной системы и стала одной из наиболее мощных универсальных интегрированных инструментов. Слово «интегрированная» указывает на то, что в этой системе объединены удобная оболочка, редактор выражений и текстовых комментариев, вычислитель и графический программный процессор. В новой версии используются такие мощные типы данных, как многомерные массивы, массивы ячеек, массивы структур, массивы Java и разреженные матрицы, что открывает возможности применения системы при создании и отладке новых алгоритмов матричных и основанных на них параллельных вычислений и крупных баз данных.

В целом MATLAB — это уникальная коллекция реализаций современных численных методов компьютерной математики, созданных за последние три десятка лет. Она вобрала в себя и опыт, правила и методы математических вычислений, накопленные за тысячи лет развития математики. Это сочетается с мощными средствами графической визуализации и даже анимационной графики. Систему с прилагаемой к ней обширной документацией вполне можно рассматривать как фундаментальный многотомный электронный справочник по математическому обеспечению ЭВМ — от массовых персональных компьютеров до супер-ЭВМ.

MATLAB представляет собой основу всего семейства продуктов MathWorks и является главным инструментом для решения широкого спектра научных и прикладных задач, в таких областях как:

1) Математические расчеты;

2) Разработка алгоритмов;

3) Моделирование;

4) Анализ данных и визуализация;

5) Научная и инженерная графика;

6) Разработка приложений, включая графический интерфейс пользователя.

MATLAB развивался в течение нескольких лет, ориентируясь на различных пользователей. В университетской среде, он представлял собой стандартный инструмент для работы в различных областях математики, машиностроении и науки. В промышленности, MATLAB – это инструмент для высокопродуктивных исследований, разработок и анализа данных.

В MATLAB важная роль отводится специализированным группам программ, называемых Toolboxes. Они очень важны для большинства пользователей MATLAB, так как позволяют изучать и применять специализированные методы. Это своего рода пакет расширения, позволяющее быстро ориентировать систему на решение задач в той или иной предметной области: в специальных разделах математики, в физике и в астрономии, в области нейронных сетей и средств телекоммуникаций, а так же в математическом моделировании.

Этот пакет подходит для анализа многокритериальных экономических задач линейного программирования. Тщательно рассмотрим систему MATLAB и ее пакеты расширения.

Основные компоненты MATLAB

Система MATLAB состоит из пяти основных частей:

1. Язык MATLAB. Это язык матриц и массивов высокого уровня с управлением потоками, функциями, структурами данных, вводом-выводом и особенностями объектно-ориентированного программирования. Это позволяет как программировать в “небольшом масштабе” для быстрого создания черновых программ, так и в “большом” для создания больших и сложных приложений.

2. Среда MATLAB. Это набор инструментов и приспособлений, с которыми работает пользователь или программист MATLAB. Она включает в себя средства для управления переменными в рабочем пространстве MATLAB, вводом и выводом данных, а также создания, контроля и отладки М-файлов и приложений MATLAB.

3. Управляемая графика. Это графическая система MATLAB, которая включает в себя команды высокого уровня для визуализации двух- и трехмерных данных, обработки изображений, анимации и иллюстрированной графики. Она также включает в себя команды низкого уровня, позволяющие полностью редактировать внешний вид графики, также как при создании Графического Пользовательского Интерфейса (GUI) для MATLAB приложений.

4. Библиотека математических функций. Это обширная коллекция вычислительных алгоритмов от элементарных функций, таких как сумма, синус, косинус, комплексная арифметика, до более сложных, таких как обращение матриц, нахождение собственных значений, функции Бесселя, быстрое преобразование Фурье.

5. Программный интерфейс. Это библиотека, которая позволяет писать программы на Си и Фортране, которые взаимодействуют с MATLAB. Она включает средства для вызова программ из MATLAB(динамическая связь), вызывая MATLAB как вычислительный инструмент и для чтения-записи MAT-файлов.

Toolboxes и его виды. Toolboxes – это дополнительные специализированные пакеты MATLAB.

Виды Toolboxes:

1. Communications Toolbox – это пакет расширения MATLAB, содержащий набор типовых функций для проектирования систем связи. Пакет включает графические приложения, алгоритмы, функции командной строки и средства визуализации для всестороннего анализа, проектирования и разработки коммуникационных систем на физическом уровне детализации.

2. Control System Toolbox – это пакет расширения для анализа, проектирования и разработки систем автоматического управления. Включает в себя всевозможные функции и графические приложения для работы с динамическими объектами и линейными замкнутыми системами управления.

3. Financial Toolbox – этот пакет дополняет возможности Statistics Toolbox и Optimization Toolbox функциями и процедурами анализа финансовых данных. Пакет служит инструментом для решения задач оптимизации портфеля инвестиций, оценки риска, анализа процентных ставок и производных ценных бумаг, а также для анализа финансовых временных рядов.

4. System Identification Toolbox содержит инструменты создания математических моделей динамических систем на основе наблюдаемых входных/выходных данных. Пакет снабжен гибким графическим интерфейсом, помогающим организовывать данные и создавать модели. Методы идентификации, входящие в пакет, применимы для широкого класс задач от проектирования систем управления и обработки сигналов до анализа временных рядов и вибраций.

5. Fuzzy Logic Toolbox – содержит инструменты для проектирования систем нечеткой логики. Пакет позволят создавать экспертные системы на основе нечеткой логики, проводить кластеризацию нечеткими алгоритмами, а также проектировать нечеткие нейросети.

6. Higher-Order Spectral Analysis содержит специальные алгоритмы для анализа сигналов с использованием моментов высшего порядка. Пакет предоставляет широкие возможности для анализа негауссовых сигналов, так как содержит алгоритмы, пожалуй, самых передовых методов для анализа и обработки сигналов.

7. Image Processing Toolbox содержит полный набор типовых эталонных алгоритмов для обработки и анализа изображений, в том числе функций фильтрации, частотного анализа, улучшения изображений, морфологического анализа и распознавания.

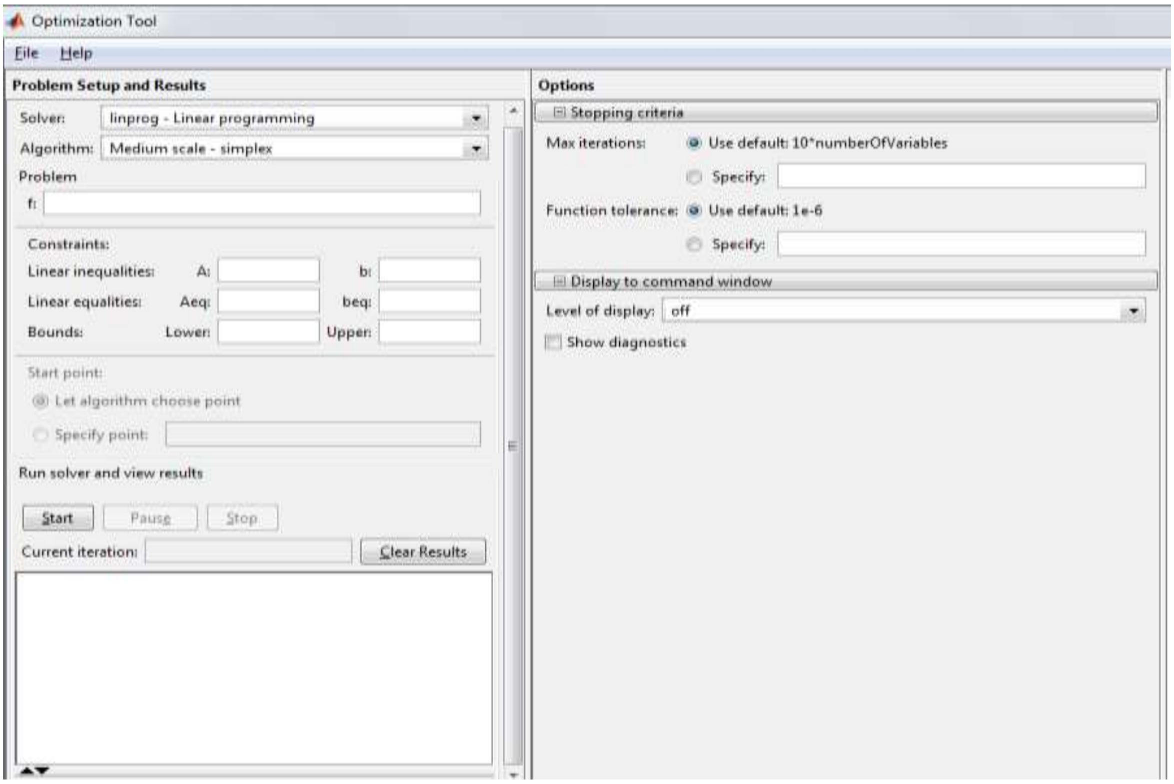
8. Mapping Toolbox содержит полный набор средств для построения карт, обработки и визуализации картографических данных. Пакет включает стандартные функции преобразования картографических проекций, расчета линий прямой видимости и другие вычисления геометрии земли, использующиеся в геодезии, картографии, экологии, океанографии, в нефтегазоразведывательной и аэрокосмической отраслях.

9. Model Predictive Control Toolbox – это пакет расширения MATLAB для исследования и проектирования алгоритмов управления с предсказанием динамики. Позволяет создавать системы адаптивного управления для сложных систем с одним или несколькими входами (выходами) и различными ограничениями.

10. Neural Network Toolbox содержит средства для проектирования, моделирования, разработки и визуализации нейронных сетей. Нейросетевые технологии позволяют решать такие задачи, решение которых классическими формальными методами затруднено или не возможно. Пакет обеспечивает всестороннюю поддержку типовых нейросетевых парадигм и имеет открытую модульную архитектуру.

11. Optimization Toolbox – это пакет расширения MATLAB, содержащий набор классических алгоритмов для оптимизации стандартных задач и задач большой размерности.

12. Genetic Algorithm - это пакет, расширяющий оптимизационные возможности MATLAB и Optimization Toolbox для решения задач



оптимизации недифференцируемых, стохастических и разрывных функций.

Для решения ряда больших (крупных) и экономически важных задач в бизнесе и в инженерных разработках выгодно применять генетические алгоритмы. Для реализации таких генетических алгоритмов не нужно создавать отдельный программный продукт, так как ряд базовых свойств этих алгоритмов остается постоянной при решении совершенно разных задач

Вернемся к основному окну утилиты Optimization Tool. В поле Number of variables (число переменных) указывается длина входного вектора оптимизируемой функции. В рассмотренном выше примере функция my_fun имеет входной вектор длины 5.

В панели Constraints (ограничения) можно задать ограничения или ограничивающую нелинейную функцию. В поле Linear inequalities задается линейное ограничение неравенством вида: $A*x \leq b$.

В поле Linear equalities данной панели задаются линейные ограничения равенством: A*x = b.

В обоих случаях A – некоторая матрица, b – вектор.

В поле Bounds (границы) в векторном виде задаются нижнее и верхнее ограничения переменных.

Если в конкретной задаче не требуется задание ограничений, все поля панели Constraints следует оставить незаполненными.

В Options, находится панель настройки графиков. Она позволяет выводить различные графики, отображающие информацию о работе генетического алгоритма. На основе этой информации можно менять настройки алгоритма с целью повышения эффективности его работы.

Подробнее эта панель будет описана ниже наравне с остальными панелями вкладки Options утилиты gamultiobj.

Панель Run Solver содержит управляющие элементы (кнопки Start, Pause и Stop для начала, временной и полной остановки работы генетического алгоритма). Также она содержит поля Current Iteration, в которое выводятся текущие результаты работы запущенного генетического алгоритма, и Final point, в котором выводится значение конечной точки работы алгоритма — наилучшей величины оптимизируемой функции (то есть, искомое значение). В правой части основного окна утилиты Optimization Tool находится панель Options. Она позволяет устанавливать различные настройки для работы генетических алгоритмов.

В среде MATLAB задачи линейного программирования решаются с помощью функции linprog. Функция linprog решает задачу линейного программирования в форме

$$\begin{aligned} &f^T x \to \inf \\ &Ax \le b, \\ &Aeqx = beqx, \\ &lb \le x \le ub. \end{aligned} \qquad (1)$$

Основными входными данными linprog являются: вектор коэффициентов целевой функции f, матрица ограничений-неравенств A, вектор правых частей ограничений-неравенств b, матрица ограничений-равенств Aeq, вектор правых частей ограничений-равенств beq, вектор lb, ограничивающий план x снизу, вектор ub, ограничивающий план x сверху. На выходе функция linprog даёт оптимальный план x задачи (1) и экстремальное значение целевой функции fval.

ПРИМЕР. Решим в MATLAB задачу линейного программирования

$$f(x) = 3x_1 + x_2 + 2x_3 \to \inf,$$
$$x_1 + x_2 + x_3 \geq 1,$$
$$2x_1 + x_2 - x_3 \geq -1,$$
$$x_1 - x_2 + x_3 = 0,$$
$$0 \leq x_1 \leq 1,$$
$$0 \leq x_2 \leq 1,$$
$$0 \leq x_3 \leq 1.$$

Соответствующая программа (m-файл)* выглядит так:

```
clear all
close all
clc % удаляются все текущие переменные из памяти MATLAB,
    закрываются все графические окна, очищается экран консоли
C = [3 1 2]; % задаётся вектор длины три
D = [1 1 1; 2 1 -1]; % строки матрицы разделяются точкой с
    запятой
B = [1 -1];
Aeq = [1 -1 1];
beq = [0];
lb = zeros(3,1); % задаётся нулевой вектор длины три
ub = [1 1 1];
f = C;
A = -D;
b = -B; % появляются знаки «-», так как ограничения-неравенства
    Dx ⩾ B приводятся к виду -Dx ⩽ -B
[x,fval] = linprog(f,A,b,Aeq,beq,lb,ub);
x
fval
```

Запустив программу, получим сообщение

```
Optimization terminated.
x =
  0
  0.5000
  0.5000
fval =
  1.5000
```

Дополнительно можно задать начальное приближение x0:

[x,fval] = linprog(f,A,b,Aeq,beq,lb,ub,x0).

Если какой-то из входных параметров отсутствует, на его место следует поставить квадратные скобки [], за исключением случая, когда это последний параметр в списке. Например, если нужно решить задачу без ограничений равенств, в которой не задано начальное приближение, то оператор вызова функции linprog будет выглядеть так:

[x,fval] = linprog(f,A,b,[],[],lb,ub).

(Квадратные скобки в конце списка, соответствующие начальному приближению, не ставятся.) С помощью входного параметра options устанавливаются некоторые дополнительные настройки, в частности, выбирается алгоритм решения.

MATLAB решает задачи линейного программирования двумя способами:

алгоритмом внутренней точки (Large-Scale Algorithm) и вариантом симплекс-метода (Medium-Scale Algorithm). По умолчанию используется алгоритм внутренней точки. Чтобы выбрать симплекс-метод, нужно написать

options = optimset('LargeScale','off','Simplex','on');

[x,fval] = linprog(f,A,b,Aeq,beq,lb,ub,[],options).

Разберёмся с выходными данными. MATLAB позволяет выводить информацию о том, как завершилось решение задачи. За это отвечает параметр exitflag. Если значение exitflag равно 1, то найдено решение задачи, если равно 0, то превышено допустимое число итераций, если равно −2 _ множество планов задачи пусто, если равно −3 _ целевая функция не ограничена снизу на множестве планов. Интерпретация других значений параметра exitflag приведена в MATLAB Help. Для симплекс-метода допустимое число итераций (MaxIter) по умолчанию в 10 раз больше количества переменных.

Значение MaxIter можно изменить. Чтобы установить допустимое число итераций равным, к примеру, 10, нужно написать

options =optimset('LargeScale','off','Simplex','on','MaxIter',10);

[x,fval] = linprog(f,A,b,Aeq,beq,lb,ub,[],options).

Если после выполнения десятой итерации решение не будет найдено, параметр exitflag станет нулевым и на экране появится сообщение

Maximum number of iterations exceeded;

increase options.MaxIter.

Параметр output содержит информацию о процессе оптимизации, в частности, число итераций (iterations) и используемый алгоритм (algorithm).

Другие поля параметра output описаны в MATLAB Help. Запустим с данными из примера 1 следующую программу:

options = optimset('LargeScale','off','Simplex','on');

[x,fval,exitflag,output] =linprog(f,A,b,Aeq,beq,lb,ub,[],options);

exitflag

output.iterations

output.algorithm

На выходе получим:

Optimization terminated.

exitflag =

1

ans =

1

ans =

medium scale: simplex

Это означает, что симплекс-метод успешно завершил работу, для нахождения решения потребовалось одна итерация.

Наконец, в выходном параметре lambda содержится решение двойственной задачи линейного программирования. Параметр lambda состоит из четырёх массивов: lambda.ineqlin, lambda.eqlin, lambda.upper, lambda.lower.

В этих массивах находятся двойственные переменные, приписанные ограничениям-неравенствам, ограничениям-равенствам, ограничениям на план сверху и снизу соответственно.

ЗАКЛЮЧЕНИЕ

Известно, что многие экономические объекты при формализации с целью моделирования функционирования допускают применение оптимизационных методов. При этом, широкий класс таких методов укладывается в рамки линейно-программных задач. Такие задачи на формальном уровне состоят в минимизации или максимизации целевой функции при линейных ограничениях.

Любая реальная экономическая задача не исчерпывается одним критерием и при планировании производственных процессов на предприятии необходимо постоянно принимать непростые решения, связанные с учетом многих критериев качества и ограничений на ресурсы.

В дипломной работе исследовались методы многокритериальной оптимизации, современные программные средства поддержки принятия решений MATLAB, а также формулирование экономическую модель в электросети и применения методов решения и пакета оптимизации к этой модели.

Для достижения цели дипломной работы изучались следующие задачи:

- рассмотрения основные понятия, принципы, структуру и особенности многокритериальной оптимизации линейного программирования;

- исследование систему производства на исследуемом предприятии;

- оптимизировать задачу о производстве на основе изученных методов и с помощью инструментального пакета MATLAB;

Информационной базой исследования явились положения и концепции, представленные в работах отечественных и зарубежных авторов.

СПИСОК ЛИТЕРАТУРЫ

ПРИЛОЖЕНИЯ