ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение высшего профессионального образования
«**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**»

Утверждаю
Декан АВТФ

С.А. Гайворонский
“_____” ______________ 2008 г.

В.И. Рейзлин

Методические указания к выполнению лабораторных работ по дисциплине «Методы оптимизации»

Издательство
Томского политехнического университета
2008

УДК 519.6
Р 22

Составитель: доцент каф. ИПС ИК В.И. Рейзлин

Рецензент: кандидат технических наук,
доцент кафедры ВТ ИК Е.А. Мирошниченко

Л А Б О Р А Т О Р Н А Я   Р А Б О Т А № 1

МЕТОДЫ ОДНОМЕРНОЙ ОПТИМИЗАЦИИ

Цель работы: изучение методов одномерного поиска, а также исследование влияния параметров алгоритмов соответствующих методов на их эффективность.

1. Описание работы

Для решения задачи оптимизации, в которой характеристическая мера задана функцией одной переменной, можно использовать различные методы. Выбор метода решения задачи оптимизации зависит от различных предположений и допущений относительно природы и свойств исследуемой функции. Ниже рассмотрены некоторые из известных методов одномерной оптимизации.

1.1. Методы исключения интервалов

Эти методы ориентированы на нахождение точки оптимума внутри заданного интервала и позволяют определить оптимум функции одной переменной путем последовательного исключения подынтервалов и, следовательно, путем уменьшения интервала поиска /1,2/.

Для того чтобы начать поиск с помощью этих методов, необходимо установить границы интервала, содержащего точку оптимума. После этого можно применить процедуру уменьшения интервала поиска с целью получения уточненных оценок координат оптимума. Величина подынтервала, исключаемого на каждом шаге, зависит от расположения пробных точек x_1 и x_2 внутри интервала поиска. Поскольку местонахождение точки оптимума априори неизвестно, целесообразно предположить, что размещение пробных точек должно обеспечивать уменьшение интервала в одном и том же отношении. Кроме того, в целях повышения эффективности алгоритма необходимо потребовать, чтобы указанное отношение было максимальным. Подобную стратегию иногда называют минимаксной стратегией поиска.

1.1.1. Метод деления интервала пополам

Рассматриваемый метод позволяет исключать в точности половину интервала на каждой итерации. Иногда этот метод называют трехточечным поиском на равных интервалах, поскольку его реализация основана на выборе трех пробных точек, равномерно распределенных в интервале поиска. Ниже приводится алгоритм поиска, ориентированный на нахождение точки минимума функции $f(x)$ в интервале (a, b).

Шаг 1. Положить $xm=(a+b)/2$ и $L=b-a$. Вычислить значение $f(xm)$.

Шаг 2. Положить $x1=a+L/4$ и $x2=b-L/4$. Таким образом, точки $x1$, $x2$ и xm делят интервал (a, b) на четыре равные части. Вычислить значения $f(x1)$ и $f(x2)$.

Шаг 3. Сравнить $f(x1)$ и $f(xm)$.
Если $f(x1)<f(xm)$, исключить интервал (xm, b), положив $b=xm$. Средней точкой нового интервала поиска становится точка x1. Следовательно, необходимо положить $xm=x1$. Перейти к шагу 5.Если $f(x1)\geq f(xm)$, перейти к шагу 4.

Шаг 4. Сравнить $f(x2)$ и $f(xm)$.
Если $f(x2)<f(xm)$, исключить интервал (a, xm), положив $a=xm$. Так как средней точкой нового интервала становится точка $x2$, положить $xm=x2$. Перейти к шагу 5.Если $f(x2)\geq f(xm)$, исключить интервалы $(a, x1)$ и $(x2, b)$. Положить $a=x1$ и $b=x2$. Средней точкой нового интервала продолжает оставаться xm. Перейти к шагу 5.

Шаг 5. Вычислить $L=b-a$. Если величина $|L|$ мала, закончить поиск. В противном случае вернуться к шагу 2.

1.1.2. Метод золотого сечения

В отличие от приведенного выше в методе золотого сечения на каждой итерации вычисляется только одно значение целевой функции. Приведем особенности реализации этого метода.

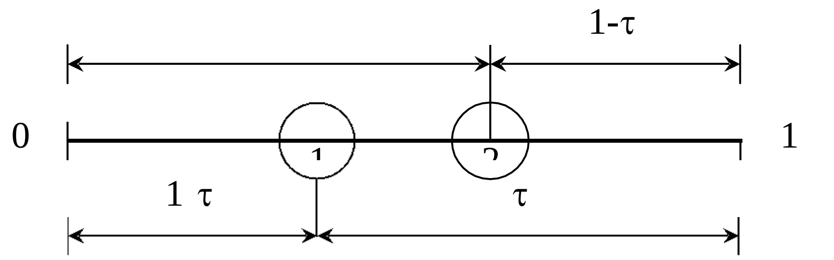

Рис 1.1. Поиск с помощью метода золотого сечения

Рассмотрим симметричное расположение двух пробных точек на исходном интервале единичной длины, которое показано на рис 1.1. Пробные точки отстоят от граничных точек интервала на расстоянии τ. При таком симметричном расположении точек длина остающегося после исключения интервала всегда равна τ независимо от того, какие из значений функции в пробных точках оказались меньшим. Предположим, что исключается правый подынтервал. На рис 1.2 показано, что оставшийся подынтервал длины τ содержит одну пробную точку, расположенную на расстоянии (1-τ) от левой граничной точки.

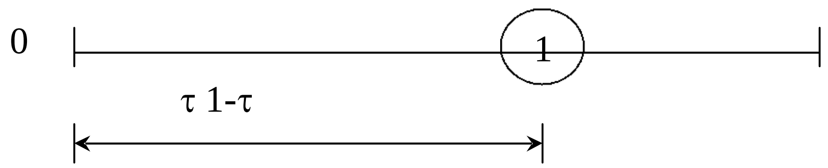

Рис 1.2. Интервалы, полученные методом золотого сечения

Для того чтобы симметрия поискового образца сохранялась, расстояние (1-τ) должно составлять τ-ю часть длины интервала (которая равна τ). При таком выборе τ следующая пробная точка размещается на расстоянии, равном τ-й части длины интервала, от правой граничной точки интервала (рис 1.3).

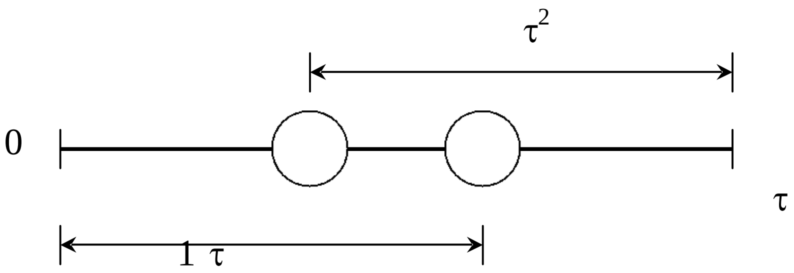

Рис 1.3. Симметрия золотого сечения интервала

Отсюда следует, что при выборе τ в соответствии с условием $1-\tau=\tau^2$ поискового образца, показанного на рис 1.1, сохраняется при переходе к уменьшенному интервалу, который изображен на рис 1.3. Решая это квадратное уравнение, получаем

$$\tau = (-1 \pm \sqrt{5})/2,$$

откуда положительное решение τ=0.61803…. Схема поиска, при которой пробные точки делят интервал в этом отношении, известна под названием поиска с помощью метода золотого сечения. Алгоритм поиска с помощью данного метода выглядит следующим образом.

Шаг 1. Положить *x0=a, x3=b*.

Шаг 2. Положить *x1=a+t1·(b-a)*. Вычислить значение *f(x1)*.

Шаг 3. Положить *x2=a+t2·(b-a)*. Вычислить значение *f(x2)*.

Шаг 4. Сравнить *f(x2)* и *f(x1)*. Если *f(x1)>f(x2)*, исключить интервал *(x0, x1)*, положив *L=x3-x1, x0=x1, x1=x2, x2=x0+t2·L, f(x1)=f(x2)*. Вычислить значение *f(x2)*. Перейти к шагу 5. Если *f(x1)<f(x2)*, исключить интервал *(x2, x3)*, положив *L=x2-x0, x3=x2, x2=x1, x1=x0+t1·L, f(x2)=f(x1)*. Вычислить значение *f(x1)*. Перейти к шагу 5.

Шаг 5. Определить количество вычислений функции - *k*. Если *k<N* и *|L|>E*, перейти к шагу 4. В противном случае закончить поиск.

1.1.3. Метод Фибоначчи

Метод Фибоначчи имеет такую же последовательность исключения интервалов, что и метод золотого сечения. Отличие состоит в выборе начальных точек и в величине исключаемого интервала. Начальная точка *x1* в методе Фибоначчи вычисляется по формуле /1/.

$$x1=((b-a)\cdot F(N-1)+E\cdot(-1)^N)/F(N)+a, \qquad (1.1)$$

где *F(i)* – числа Фибоначчи, *i=0,1,2,…,N*;

N – число вычислений функции;

E – заданная точность.

Как видно из формулы *(1.1),* для того чтобы получить начальную точку необходимо заранее знать число вычислений функции, за которое будет в результате поиска получен оптимум с заданной точностью. Ниже приводится алгоритм поиска для данного метода.

Шаг 1. Положить $x0=a$, $x3=b$. Вычислить значение $x1$ по формуле (1.1). Вычислить значение $f(x1)$.

Шаг 2. Положить $x2=x0-x1+x3$. Вычислить значение $f(x2)$.

Шаг 3. Сравнить $f(x2)$ и $f(x1)$.

Если $f(x2)>f(x1)$, сравнить $x1$ и $x2$.Если $x2>x1$, исключить интервал $(x2, x3)$, положив $x3=x2$. Перейти к шагу 4. Если $x2<x1$, исключить интервал $(x0, x2)$, положив $x0=x2$. Перейти к шагу 4. Если $f(x2)<f(x1)$, сравнить $x1$ и $x2$. Если $x1>x2$, исключить интервал $(x1, x3)$, положив $x3=x1$, $x1=x2$, $f(x1)=f(x2)$. Перейти к шагу 4. Если $x1<x2$, исключить интервал $(x0, x1)$, положив $x0=x1$, $x1=x2$, $f(x1)=f(x2)$. Перейти к шагу 4.

Шаг 4. Определить количество вычислений функции - k. Если $k<N$ перейти к шагу 2. В противном случае закончить поиск.

1.2. Полиномиальная аппроксимация и методы точечного оценивания

Основная идея рассматриваемых методов связана с возможностью аппроксимации гладкой функции полиномом и последующего использования аппроксимирующего полинома для оценивания координаты точки оптимума /1,2/. Необходимыми условиями эффективной реализации такого подхода являются унимодальность и непрерывность исследуемой функции. Согласно теореме Вейерштрасса об аппроксимации, если функция непрерывна в некотором интервале, то ее с любой степенью точности можно аппроксимировать полиномом достаточно высокого порядка. Следовательно, если функция унимодальна и найден полином, который достаточно точно ее аппроксимирует, то координату точки оптимума функции можно оценить путем вычисления координаты точки оптимума полинома. Согласно теореме Вейерштрасса, качество оценок координаты точки оптимума, получаемых с помощью аппроксимирующего полинома, можно повысить двумя способами: использованием полинома более высокого порядка и уменьшением интервала аппроксимации. Второй способ, вообще говоря, является более предпочтительным, поскольку построение аппроксимирующего полинома порядка выше третьего становиться весьма сложной процедурой, тогда как уменьшение интервала в условиях, когда выполняется предположение об унимодальности функции, особой сложности не представляет.

1.2.1. Метод оценивания с использованием квадратичной аппроксимации

Простейшим вариантом полиномиальной интерполяции является квадратичная аппроксимация, которая основана на том факте, что функция, принимающая минимальное значение во внутренней точке интервала, должна быть, по крайней мере, квадратичной. Если же функция линейная, то ее оптимальное значение может достигаться только в одной из двух граничных точек интервала. Таким образом, при реализации метода оценивания с использованием квадратичной аппроксимации предполагается, что в ограниченном интервале можно аппроксимировать функцию квадратичным полиномом, а затем использовать построенную аппроксимирующую схему для оценивания координаты точки истинного минимума функции.

Если задана последовательность точек x_1, x_2, x_3 и известны соответствующие этим точкам значения функции f_1, f_2, f_3, то можно определить постоянные величины a_0, a_1 и a_2 таким образом, что значения квадратичной функции

$$q(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2)$$

совпадут со значениями $f(x)$ в трех указанных точках. Перейдем к вычислению $q(x)$ в каждой из трех заданных точек. Прежде всего, так как

$$f_1 = f(x_1) \quad q(x_1) = a_0,$$

имеем $a_0 = f_1$.
Далее, поскольку

$$f_2 \quad f(x_2) = q(x_2) = f_1 + a_1(x_2 \quad x_1),$$

получаем

$$a_1 = (f_2 - f_1)/(x_2 - x_1). \qquad (1.2)$$

Наконец, при $x=x_3$

$$f_3 = f(x_3) = q(x_3) = f_1 + (f_2 - f_1)/(x_2 - x_1)(x_3 - x_1) + a_2(x_3 - x_1)(x_3 - x_2).$$

Разрешая последнее уравнение относительно a_2, получаем

$$a_2 = \frac{1}{x_3 - x_2}\left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1}\right). \qquad (1.3)$$

Таким образом, по трем заданным точкам и соответствующим значениям функции можно оценить параметры a_0, a_1 и a_2 аппроксимирующего квадратичного полинома с помощью приведенных выше формул.

Если точность аппроксимации исследуемой функции в интервале от x_1 до x_3 с помощью квадратичного полинома оказывается достаточно высокой, то в соответствии с предложенной стратегией поиска построенный полином можно использовать для оценивания координаты точки оптимума. Стационарные точки функции одной переменной определяются путем приравнивания к нулю ее первой производной и последующего нахождения корней полученного таким образом уравнения. В данном случае из уравнения

$$\frac{dq}{dx}=a_1+a_2(x-x_2)+a_2(x-x_1)=0$$

можно получить

$$\bar{x}=(x_2+x_1)/2-(a_1/2\cdot a_2). \qquad (1.4)$$

Поскольку функция *f(x)* на рассматриваемом интервале обладает свойством унимодальности, а аппроксимирующий квадратичный полином также является унимодальной функцией, то можно ожидать, что величина $\bar{x}$ окажется приемлемой оценкой координаты точки истинного оптимума x^* /2/. Схема алгоритма получения оптимальной точки можно описать следующим образом.

Шаг 1. Положить *x1=a, x3=b, x2=(x3-x1)/2.*

Шаг 2. Вычислить *f(x1), f(x2), f(x3).*

Шаг 3. По трем точкам *x1, x2, x3* вычислить параметры *a1, a2,* используя формулы (1.2) и (1.3), то есть

$$a1=\frac{f(x2)\quad f(x1)}{x2-x1},$$

$$a2=\frac{1}{(x1)}\left(\frac{f(x3)-f(x1)}{x3-x2}-\frac{f(x2)-f}{x3-x1}\right).$$

$$x2-x1$$

Шаг 4. Вычислить оптимум $\bar{x}$ с помощью параметров *a1* и *a2,* используя формулу (1.4).

1.2.2. Метод Пауэлла

Этот метод, разработанный Пауэллом, основан на последовательном применении процедуры оценивания с использованием квадратичной аппроксимации. Схему алгоритма можно описать следующим образом. Пусть *x1* - начальная точка, *Δx* – выбранная величина шага по оси *x.*

Шаг 1. Вычислить *x2=x1+Δx.*

Шаг 2. Вычислить *f(x1)* и *f(x2).*

Шаг 3. Если *f(x1)>f(x2)*, положить *x3=x1+2Δx*. Если *f(x1)≤f(x2)*, положить *x3=x1-Δx*.

Шаг 4. Вычислить *f(x3)* и найти

Fmin=min{f1, f2, f3},

Xmin= точка *xi*, которая соответствует *Fmin*.

Шаг 5. По трем точкам *x1, x2, x3* вычислить $\bar{x}$, используя формулы (1.2), (1.3), (1.4).

Шаг 6. Проверка на окончание поиска. Является ли разность *Fmin* - $f(\bar{x})$ достаточно малой? Является ли разность *Xmin* - $\bar{x}$ достаточно малой? Если оба условия выполняются, закончить поиск. В противном случае перейти к шагу 7.

Шаг 7. Выбрать «наилучшую» точку *(Xmin или* $\bar{x}$*)* и две точки по обе стороны от нее. Обозначить эти точки в естественном порядке и перейти к шагу 4.

При первой реализации шага 5 границы интервала, содержащего точку минимума, не обязательно оказываются установленными. При этом полученная точка $\bar{x}$ может находиться за точкой *x3*. Для того чтобы исключить возможность слишком большого экстраполяционного перемещения, следует провести после шага 5 дополнительную проверку и в случае, когда точка $\bar{x}$ находится слишком далеко от *x3*, заменить $\bar{x}$ точкой, координата которой вычисляется с учетом заранее установленной длины шага.

1.3. Методы с использованием производных

Все рассмотренные в предыдущих разделах методы поиска основываются на предположениях об унимодальности и в ряде случаев о непрерывности исследуемой целевой функции. Целесообразно предположить, что если в дополнение к условию непрерывности ввести требование дифференцируемости функции, то эффективность поисковых процедур можно существенно повысить. Напимним, что необходимым условием существования локального минимума функции в некоторой точке *z* является обращение в нуль первой производной функции в этой точке, т. е. $f\,(z) = df\,/\,dx\,|_{x=z} = 0$.

Если функция *f(x)* содержит члены, включающие *x* в третьей и более высоких степенях, то непосредственное получение аналитического решения уравнения *f?(x)=0* может оказаться затруднительным. В таких случаях используются приближенные методы последовательного поиска стационарной точки функции *f*.

1.3.1. Метод Ньютона-Рафсона

В рамках схемы Ньютона-Рафсона предполагается, что функция *f* дважды дифференцируема. Работа алгоритма начинается в точке x_1, которая представляет начальное приближение (или начальную оценку) координаты стационарной точки или корня уравнения *f?(x)=0*. Затем строится линейная аппроксимация функции $f?(x)$ в точке x_1, и точка, в которой аппроксимирующая линейная функция обращается в нуль, принимается в качестве следующего приближения. Если точка $x_к$ принята в качестве текущего приближения к стационарной точке, то линейная функция, аппроксимирующая функцию $f?(x)$ в точке $x_к$, записывается в виде

$$\tilde{f}'(x;x_k) = f'(x_k) + f''(x_k)(x - x_k). \qquad (1.5)$$

Приравняв правую часть уравнения (1.5) нулю, получим следующее приближение:

$$x_{k+1} \quad x_k - [f'(x_k)/ f'(x_k)]. \qquad (1.6)$$

Рисунок 1.4 иллюстрирует основные шаги реализации метода Ньютона. К сожалению, в зависимости от выбора начальной точки и вида функции алгоритм может как сходиться к истинной стационарной точке, так и расходиться /2/.

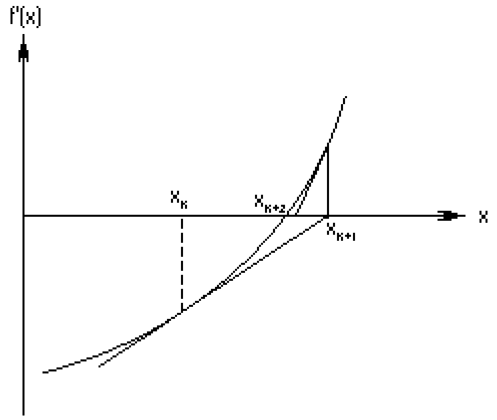

Рисунок 1.4 - Метод Ньютона Рафсона

Ниже приведен алгоритм, использующий поиск методом Ньютона-Рафсона.

Шаг 1. Определить начальную точку $x1$.

Шаг 2. Вычислить $f?(x_k)$, $f''(x_k)$ и следующее приближение по формуле (1.6).

Шаг 3. Проверка на окончание поиска. Если $| f?(x_k)|>E$, то перейти к шагу 2.

1.3.2. Метод средней точки

Если функция $f?(x)$ унимодальна в заданном интервале поиска, то точкой оптимума является точка, в которой $f?(x)=0$. Если при этом имеется возможность вычислять как значение функции, так и ее производной, то для нахождения корня уравнения $f?(x)=0$ можно воспользоваться эффективным алгоритмом исключения интервалов, на каждой итерации которого рассматривается лишь одна пробная точка. Например, если в точке z выполняется неравенство $f?(z)<0$, то с учетом предположения об унимодальности естественно утверждать, что точка минимума не может находиться левее точки z. Другими словами, интервал $x \le z$ подлежит исключению. С другой стороны, если $f'(z)>0$, то точка минимума не может находиться правее z и интервал $x \ge z$ можно исключить /2/. Приведенные рассуждения лежат в основе логической структуры метода средней точки, который иногда называют поиском Больцано.

Определим две точки L и R таким образом, что $f'(L)<0$ и $f'(R)>0$. Стационарная точка расположена между L и R. Вычислим значение производной функции в средней точке рассматриваемого интервала $z=(L+R)/2$. Если $f'(z)>0$, то интервал (z, R) можно исключить из интервала поиска. С другой стороны, если $f'(z)<0$, то можно исключить интервал (L, z). Ниже дается формализованное описание шагов алгоритма.

Шаг 1. Положить $R=b$, $L=a$; при этом $f'(a)<0$ и $f'(b)>0$.

Шаг 2. Вычислить $z=(R+L)/2$ и $f'(z)$.

Шаг 3.Если $|f'(z)| \le E$, закончить поиск. В противном случае, если $f'(z)<0$, положить $L=z$ и перейти к шагу 2. Если $f'(z)>0$, положить $R=z$ и перейти к шагу 2.

Следует отметить, что логическая структура поиска в соответствии с изложенным методом исключения интервалов основана лишь на исследовании знака производной независимо от значений, которые эта производная принимает.

1.3.3. Метод секущих

Метод секущих, являющийся комбинацией метода Ньютона и общей схемы исключения интервалов, ориентирован на нахождение корня уравнения $f'(x)$ в интервале (a, b), если, разумеется, такой корень существует.

Предположим, что в процессе поиска стационарной точки функции $f(x)$ в интервале (a, b) обнаружены две точки L и R, в которых знаки производной различны. В этом случае алгоритм метода секущих позволяет аппроксимировать функцию $f'(x)$ «секущей прямой» (прямой линией, соединяющей две точки) и найти точку, в которой секущая графика $f'(x)$ пересекает ось абсцисс (рис. 1.5). Таким образом, следующее приближение к стационарной точке x^* определяется по формуле

$$z = R - \frac{f'(R)}{[f'(R) - f'(L)]/(R-L)}. \qquad (1.7)$$

Если $|f'(z)| \le E$, поиск следует закончить. В противном случае необходимо выбрать одну из точек L или R таким образом, чтобы знаки производной в этой точке и точке z были различны, а затем повторить основной шаг алгоритма /2/.

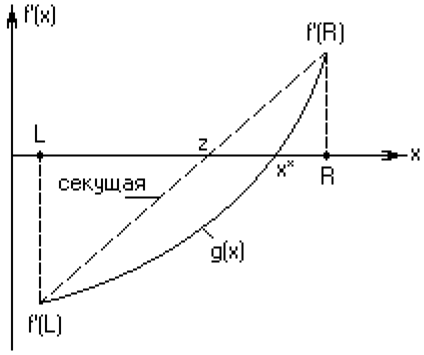

Рисунок 1.5 - Метод секущих

Алгоритм для данного метода приводится ниже.

Шаг 1. Положим $R=b$, $L=a$; при этом *f?(a)<0* и *f?(b)>0*.

Шаг 2. Вычислить текущее приближение z к минимуму по формуле (1.7). Вычислить *f?(z)*.

Шаг 3. Если | *f?(z)*|≤*E*, закончить поиск. В противном случае, если *f?(z)>0*, положить $R=z$ и перейти к шагу 2. Если *f?(z)<0*, положить $L=z$ и перейти к шагу 2.

После составления алгоритмов можно начинать разработку программного комплекса, определив вначале цели, которым должна удовлетворять разрабатываемая программа.

2. Реализация программного комплекса

Комплекс для изучения и исследования методов одномерного поиска реализован в среде MATLAB.

После запуска комплекса появляется экранная форма, содержащая определенное количество окон, с помощью которых можно задавать методы решения задачи, вариант и параметры исследования алгоритмов.

Первые три окна <Раздел>, <Метод>, <Вариант> позволяют выбрать соответственно группу методов решения задачи, в рамках заданной группы – метод решения и вариант, определяющий вид исследуемой целевой функции.

Следующие окна по отображению графиков исследуемой функции, диапазону ее исследования, виду целевой функции, числу итераций N, по поиску ее оптимума, ошибки, шагу являются активными. Пользователь при решении задачи может поменять перечисленные выше параметры, параметры целевой функции и осуществлять исследования индивидуальных пользовательских функций, используя при этом пошаговый или автоматический режимы. В пошаговом режиме пользователь последовательно осуществляет обращение к виртуальной клавише «Выполнить».

Результаты исследований представляются пользователю в графическом виде и в виде последовательности итераций с соответствующими количественными оценками, которые выводятся в определенных окнах экранной формы.

Справочная информация, поясняющая особенности реализации каждого из методов, также выводится в соответствующем окне экранной формы.

3. Выполнение работы

1. Ознакомиться с работой программного комплекса.
2. Найти решение тестовой задачи и сравнить с предварительно полученным решением.
3. Получить варианты заданий на выполнение работы.
4. Исследовать влияние параметров заданных целевых функций на эффективность получаемых решений.
5. Исследовать влияние задаваемой точности на показатели быстродействия методов.
6. Оформить отчет с результатами исследования методов.

Отчет должен содержать цель работы, результаты решения заданного варианта целевой функции, дополненные соответствующими аналитическими расчетами. Помимо этого необходимо представить в форме, удобной для анализа, результаты исследования различных параметров на показатели эффективности методов. Привести результаты решения пользовательской функции.

Л А Б О Р А Т О Р Н А Я Р А Б О Т А № 2

МЕТОДЫ МНОГОМЕРНОЙ ОПТИМИЗАЦИИ

Цель работы: изучение особенностей решения задач оптимизации с использованием методов многомерной оптимизации при отсутствии ограничений.

1. **Основные положения**

Для нахождения экстремума целевых функций многих переменных можно использовать различные методы /1, 2, 3, 4/. В зависимости от особенностей организации поиска экстремума методы многомерной оптимизации можно разделить на две группы: методы, использующие собственно значения целевых функций, и методы с использованием производных. В работе рассматриваются методы второй группы. В основе методов этой группы лежит использование итерационной процедуры

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} \cdot s(x^{(k)}), \qquad (1)$$

где $x^{(k)}$ – текущее значение аргумента исследуемой функции;

$\alpha^{(k)}$ – параметр, характеризующий длину шага;

$s(x^{(k)}) = s^{(k)}$ – направление поиска в N-мерном пространстве управляемых переменных.

В особенности от определения величины $\alpha^{(k)}$ и направления $s(x^{(k)})$ выделяют ряд методов оптимизации.

1.1. Градиентные методы

Градиентные методы в отличие от методов прямого поиска, которые используют в процедурах поиска только значения целевой функции в исследуемых точках, предполагают наличие информации о производных функции. Это позволяет сократить количество необходимых вычислений значений исследуемой функции.

Если в качестве направления поиска $s(x^{(k)})$ принять направление антиградиента функции

$$s(x^{(k)}) = -\nabla f(x^{(k)}) \ ,$$

где $\nabla f = (\frac{\partial f}{\partial x_1};\frac{\partial f}{\partial x_2};...;\frac{\partial f}{\partial x_n})$ - градиент функции,

то получим из (1) соотношение

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)}\nabla f(x^{(k)}), \qquad (2)$$

оторое определяет реализацию метода наискорейшего спуска, или метода Коши. В этом методе значение $\alpha^{(k)}$ на каждой итерации вычисляется путем решения задачи поиска минимума функции $f(x^{(k+1)})$ вдоль направления $\nabla f(x^{(k)})$ с помощью того или иного метода одномерного поиска.

Метод обладает высокой надежностью и устойчивостью. Однако методу свойственны и некоторые недостатки /1/.

Приняв в качестве параметра $\alpha^{(k)}$ в (1) некоторое положительное число α, получим вычислительную схему

$$x^{(k+1)} = x^{(k)} - \alpha\nabla f(x^{(k)}), \qquad (3)$$

которая определяет реализацию простейшего градиентного метода при поиске минимума многомерной функции. Метод обладает рядом недостатков, из которых следует отметить следующие: во-первых, возникает необходимость выбора подходящего значения параметра α; и во-вторых, методу свойственна медленная сходимость к точке минимума вследствие малости градиента ∇f в окрестности почти стационарной области.

Наряду с приведенными выше для решения задачи оптимизации используется метод Гаусса-Зайделя, известный еще под названием метода покоординатного спуска. Суть метода заключается в организации поиска минимума функции последовательно по каждой координате. Пусть установлена очередность изменения координат $x_1, x_2, ..., x_n$, совпадающая с очередностью их индексов 1, 2, …, n. Сначала изменяем одну координату x_1, сохраняя все остальные координаты постоянными, на некоторую величину Δx и определяем величину приращения Δf . Если $f > 0$, то, следовательно, шаг сделан в ложном направлении. Изменяем направление и движемся до тех пор, пока $f \quad 0$ не изменит знак на противоположный. Тогда переходим к изменению другой координаты x_2 и т.д. до тех пор, пока не будет найдена с некоторой заданной точностью точка минимума. При реализации метода возможны другие вычислительные схемы.

Метод отличается простотой реализации. Однако по сравнению с другими методами требует большего времени на поиск минимума функции.

1.2. Метод Ньютона

Рассмотренные выше методы основаны на последовательной линейной аппроксимации целевой функции и требуют вычисления значений целевой функции и ее производных на каждом шаге, число которых может быть очень велико /1, 3/. Для построения более общей стратегии необходимо привлечь информацию о вторых производных $f \quad x$. Эта информация может быть получена при квадратичной аппроксимации функции, когда при ее разложении в ряд Тейлора учитываются члены ряда до второго порядка включительно. Использование результатов аппроксимации приводит к реализации метода Ньютона по формуле

$$x^{(k+1)} = x^{(k)} - \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}), \qquad (4)$$

где $\nabla^2 f(x)$ - гессиан (матрица Гессе).

Метод Ньютона обнаруживает квадратичную скорость сходимости, т. е. выполняется неравенство

$$\left\| \varepsilon^{(k+1)} \right\| \le c \left\| \varepsilon^{(k)} \right\|^2,$$

где c – некоторая постоянная, связанная с обусловленностью матрицы Гессе.

$$\varepsilon^{(k)} = x^{(k)} - x^* \ ,$$

где x^* - искомое решение.

Сходимость метода Ньютона во многом зависит от выбора начального приближения $x^{(0)}$. Метод сходится всякий раз, когда выбор $x^{(0)}$ осуществляется в соответствии с условием /1/

$$\left\| \varepsilon^{(0)} \right\| < 1/c .$$

Квадратичная скорость сходимости объясняется тем обстоятельством, что при исследовании неквадратичных функций метод Ньютона не отличается высокой надежностью /1, 3/. Если точка $x^{(0)}$ находится на значительном расстоянии от точки x^*, шаг по методу Ньютона часто оказывается чрезмерно большим, что может привести к отсутствию сходимости. Метод можно довольно просто модифицировать с тем, чтобы обеспечить уменьшение целевой функции от итерации к итерации и осуществлять поиск вдоль прямой, как в методе Коши. Последовательность итераций строится в соответствии с формулой

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}) \quad . \tag{6}$$

Выбор $\alpha^{(k)}$ осуществляется таким образом, чтобы

$$f(x^{(k+1)}) \to \min .$$

Это гарантирует выполнение неравенства

$$f(x^{(k+1)}) \le f(x^{(k)}) \tag{7}$$

Такой метод носит название модифицированного метода Ньютона и в случаях, когда вычисление точных значений первых и вторых производных не сопряжено с существенными трудностями, оказывается надежным и эффективным.

1.3. Методы сопряжённых градиентов

Эти методы, обладая положительными свойствами методов Коши и Ньютона, основаны на вычисление значений только первых производных, отличаются высокой надежностью при поиске x^* из удаленной точки $x^{(0)}$ и быстро сходятся в окрестности точки минимума. В основе методов лежит процедура построения сопряженных направлений, для получения которых применяется квадратичная аппроксимация функции $f(x)$ и значения компонент градиента.

Итак, считаем, что целевая функция является квадратичной:

$$f(x) = q(x) = a + b^T x + 1/2\, x^T C x ,$$

а итерации выполняются по формуле (1), т. е.

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} \cdot s(x^{(k)}) .$$

Направления поиска на каждой итерации определяются с помощью следующих соотношений:

$$s^{(k)} = -g^{(k)} + \sum_{i=0}^{k-1} \gamma^{(i)} s^{(i)}\,, \qquad (8)$$

$$s^{(0)} = -g^{(0)}, \qquad (9)$$

где

$$g^{(k)} = \nabla f(\mathrm{x}^{(\mathrm{k})}) = \nabla \mathrm{g}(\mathrm{x}^{(\mathrm{k})}) = \mathrm{Cx}^{(\mathrm{k})} + b. \qquad (10)$$

Используя свойство квадратичной функции

$$\Delta g(x) = g(x^{(k)}) - g(x^{(k-1)}) = C(x^{(k)} - x^{(k-1)}) = C\Delta x$$

и условия С – сопряженности направлений $s^{(i)}$ (i=1, 2,…), можно получить соотношение для вычисления параметра $\gamma^{(i)}$ в формуле (8) /1, 3/, с использованием которого общая формула (8) определения направления поиска примет следующий вид:

$$s^{(k)} = -g^{(k)} + \frac{\left|g^{(k)}\right|^2}{\left\|g^{(k-1)}\right\|^2} s^{(k-1)}. \qquad (11)$$

Если $f(x$ - квадратичная функция, то для нахождения точки минимума требуется N-1 таких направлений и провести N поисков вдоль прямой. Если же функция $f(x$ не является квадратичной, количество направлений и соответствующих поисков возрастает.

1.4. Другие методы

Наряду с рассмотренными выше методами для решения задач безусловной многомерной оптимизации используются и многие другие /1, 2, 3, 6 ,7, 8/, среди которых следует отметить квазиньютоновские методы. Эти методы обладают положительными качествами метода Ньютона, однако, используют только первые производные. Во всех методах указанного класса построение векторов направлений поиска осуществляется с помощью формулы (1), в которой $s(x^{(k)})$ записывается в виде

$$s(x^{(k)}) = A^{(k)} \nabla f(x^{(k)}), \qquad (12)$$

где $A^{(k)}$ - матрица порядка N   N, которая носит название метрики. Методы поиска вдоль направлений, определяемых этой формулой, называются еще методами переменной метрики, поскольку матрица А изменяется на каждой итерации. Изменение матрицы $A^{(k)}$ связано с необходимостью приближения ее к матрице, обратной матрице Гессе. Для аппроксимации матрицы, обратной гессиану, используется рекуррентное соотношение

$$A^{(\kappa+1)} = A^{(\kappa)} + A_c^{(k)}, \qquad (13)$$

где $A_c^{(k)}$ - корректирующая матрица.

Эта матрица позволяет на каждом шаге улучшать решение задачи.

Среди квазиньютоновских методов известны метод Дэвидона-Флеттчера-Пауэлла (ДФП), метод Бройдена-Флетчера-Шэнно и др. /1, 3, 6, 7, 8/.

Приведенные выше методы обладают различными показателями эффективности. Некоторые вопросы исследования эффективности методов рассматриваются в данной лабораторной работе.

2. Реализация программного комплекса

Комплекс для изучения и исследования методов безусловной оптимизации реализован в среде MATLAB.

После запуска комплекса появляется экранная форма, содержащая несколько окон, с помощью которых можно задавать методы решения задач. Активизация окна многомерной оптимизации приводит к появлению новой экранной формы для выполнения лабораторной работы. Окна экранной формы позволяют задавать исследуемые функции, изменять ее параметры и параметры исследования алгоритмов, осуществлять вывод результатов решения задач.

Верхнее окно экранной формы позволяет выбрать вариант исследуемой функции, а в следующем окне можно изменять ее параметры. Следующие окна по заданию начального приближения, точности, числа итераций и величины шага являются активными. Исследование методов предусматривает решение задачи в пошаговом и автоматическом решении, причем результаты отображаются как графически, так и в виде количественных оценок. Можно задать и трехмерную графику.

Справочная информация, поясняющая особенности реализации каждого из методов, также выводится в соответствующем окне экранной формы.

3. Выполнение работы

1.Ознакомится с работой программного комплекса.

2.Найти решение тестовой задачи и сравнить с предварительно полученным решением.

3.Получить варианты заданий на выполнение работы.

4.Исследовать влияние параметров заданных целевых функций на эффективность решений.

5.Исследовать влияние начальных приближений, точности на показатели быстродействия методов.

6.Получить решение для пользовательской функции.

7.Оформить отчет с результатами исследования методов.

Л А Б О Р А Т О Р Н А Я Р А Б О Т А № 3

МЕТОДЫ УСЛОВНОЙ ОПТИМИЗАЦИИ

Цель работы: изучение особенностей решения задач оптимизации с использованием методов условной оптимизации.

1. Основные положения

Для нахождения экстремума целевых функций многих переменных можно использовать различные методы/1,3,5,7/. В данной работе рассматриваются методы условной оптимизации. Ограничения, накладываемые на переменные, могут иметь вид равенств и неравенств. Здесь рассмотрены методы для решения задач с ограничениями обоих видов.

1.1. Модифицированный метод Хука-Дживса

Для решения задач оптимизации с ограничениями может быть использован метод Хука-Дживса. При этом общая процедура поиска минимума целевой функции дополняется проверкой принадлежности изменяемых переменных области допустимых решений, определяемой ограничениями задачи. И если переменная выходит за допустимую область , то целевой функции присваивается некоторое большое предварительно заданное значение, что адекватно неудаче при поиске направления движения в пространстве параметров. В случае принадлежности значений переменных области допустимых решений общая процедура решения задачи не изменяется [2].

1.2. Методы множителей Лагранжа

Для нахождения экстремума целевой функции

$$f(x) = f(x_1, x_2, ..., x_n) \qquad (1)$$

при ограничениях

$$g_i(x_1, x_2, ..., x_n) = 0, \qquad i \quad 1,2,...,m. \qquad (2)$$

можно воспользоваться классическим методом условной оптимизации функций нескольких переменных /2,5/. При этом полагаем, что функции $f(x_1, x_2, ..., x_n)$ и $g_i(x_1, x_2, ..., x_n)$ непрерывны вместе со своими первыми частными производными. При этом для решения задачи составляется функция

$$F(x_1, x_2, ..., x_n, \lambda_1, \lambda_2, ..., \lambda_m) = f(x_1, x_2, ..., x_n) + \sum_{i=1}^{m} \lambda_i \cdot q_i(x_1, x_2, ..., x_n), \qquad (3)$$

определяются частные производные $\frac{F}{\partial x_j}(j=1,2,...,n)$, $\frac{\partial F}{\partial \lambda_i}(i=1,2,...,m)$ и приравниваются нулю, в результате чего получается система уравнений:

$$\begin{cases} \frac{\partial F}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g}{\partial x_j} = 0 \\ \frac{\partial F}{\partial x_j} = g_i(x_1, x_2, ..., x_n) = 0 \end{cases}, \text{ где } i \quad 1,2,...,m, \text{ а } j = 1,2,...,n \; . \qquad (4)$$

Функция (3) называется функцией Лагранжа, а числа λ_i – множителями Лагранжа. Если функция $f(x)$ $f(x_1, x_2, ..., x_n)$ в точке $X^{(0)} = (x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)})$ имеет экстремум, то существует такой вектор $\Lambda^{(0)} = (\lambda_1^{(0)}, \lambda_2^{(0)}, ..., \lambda_m^{(0)})$, что точка $(x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)}, \lambda_1^{(0)}, \lambda_2^{(0)}, ..., \lambda_m^{(0)})$ является решением системы. Следовательно, решая систему, получаем множество точек, в которых функция $f(x)$ может иметь экстремальные значения.

Метод Лагранжа можно использовать в случае ограничений в виде неравенств. Вводя дополнительные переменные, ограничения – неравенства можно преобразовать в уравнения, причем на дополнительные переменные накладываются ограничения неотрицательности.

1.3. Метод штрафных функций

Рассматриваемый метод численного решения оптимизационных задач при наличии ограничений относится к методам оптимизации на основе преобразования задачи /1, 2,3,9/.

Задача может быть сформулирована следующим образом:

минимизировать $f \; x)$, $\quad x \in R^N$ (1)

при ограничениях $g_i(x) \quad 0, j = 1,2,..m,$ (2)

$h_i(x) = 0, i = 1,2,...l.$ (3)

Суть метода заключается в преобразовании исходной целевой функции (1) путём включения в нее функции от ограничений (2) и (3), получая таким образом задачу безусловной оптимизации, для решения которой можно использовать известные методы /1,2,3/.

Преобразованная функция определяется выражением

$$P(x, R) = f(x) + \Omega(R, g \; x), h(x)), \qquad (4)$$

где Ω - штрафная функция от ограничений задачи, а R – штрафной параметр.

Существуют различные типы штрафов и различные процедуры учёта ограничений при переходе к задаче безусловной оптимизации.

1. *Квадратичный штраф*

Этот вид штрафа используется для учёта ограничений – равенств и имеет вид

$$\Omega = R(h(x))^2 \qquad . \qquad (5)$$

При минимизации этот штраф препятствует отклонению величины $h(x$ от нуля. Легко видеть, что при увеличении R стационарная точка штрафной функции P(x, R) приближается к искомому решению x^*, так как в пределе $h_k(x^{(t)}) = 0$, где t=1, 2,..., T. Функция Ω непрерывна и имеет непрерывные производные.

2. *Логарифмический штраф*

Этот и рассматриваемые далее виды штрафов учитывают ограничения – неравенства . Логарифмический штраф имеет вид

$$\Omega \quad R\ln[g(x)] \qquad (6)$$

Штраф положителен при всех x, таких, что 0 < g(x) < 1, и отрицателен при g(x) > 1. В этом случае внутренним точкам области допустимых решений отдаётся предпочтение. Логарифмический штраф - это барьерная функция, не определенная в недопустимых точках (т.е. для таких , в которых g(x) < 0). Поэтому на начальном этапе необходимо обеспечить попадание в допустимую область. Поскольку преобразованная задача решается одним из численных методов, то возможно появление недопустимых точек в процессе решения (например, как результата большого первого шага при одномерном поиске). В связи с этим должны быть предусмотрены специальные меры по предотвращению этой ситуации либо её обнаружению и устранению. Итерационный процесс начинается из допустимой точки при положительном начальном R (R=10 или R=100). После решения каждой подзадачи безусловной оптимизации уменьшается и в пределе стремится к нулю.

3. *Штраф, заданный обратной функцией*

Этот вид штрафа

$$\Omega = R[1/g \ x)] \qquad (7)$$

не имеет отрицательных значений в допустимой области. Как и предыдущий, является барьерным штрафом. В допустимых точках вблизи границы значения штрафа положительны и быстро убывают при продвижении внутрь допустимой области. На самой границе значение P (x, R) и её градиент не определены. Как и в предыдущем случае, возможно появление недопустимых точек.

4. Штраф типа квадрата срезки

$$\Omega = R \cdot \langle g(x) \rangle^2 , \qquad (8)$$

$$\text{где } \langle a \rangle = \begin{cases} a, если a \le 0, \\ 0, если a > 0. \end{cases} \qquad (9)$$

Заметим , что этот штраф внешний. Недопустимые точки не создают в данном случае сложностей по сравнению с допустимыми. Различие между ними состоит в том, что в допустимых и граничных точках штраф равен нулю. Этот вид штрафа удобен тем, что P (x, R) непрерывна и определена всюду. Вычисления производятся с положительными R, увеличивающимися от итерации к итерации.

В данной лабораторной работе рассматриваются решения задач нелинейного программирования методом штрафных функций с учётом ограничений–равенств и ограничений–неравенств.

Алгоритм решения задачи можно представить в виде следующей последовательности шагов

Шаг 1. Задать начальные данные N, J, K, $\varepsilon_1, \varepsilon_2, \varepsilon_3,$ $x^{(0)}$, $R^{(0)}$, где

ε_1 – параметр окончания одномерного поиска (если таковой используется в процедуре безусловной оптимизации);

ε_2 – параметр окончания процедуры безусловной оптимизации;

ε_3 – параметр окончания работы алгоритма;

$x^{(0)}$ – начальная точка;

$R^{(0)}$ – начальный вектор штрафных параметров.

Шаг 2. Построить штрафную функцию

$$P(x,R) \quad f(x) \quad \Omega(R, g(x), h(x)).$$

Шаг 3. Найти $x^{(t+1)}$, доставляющий экстремум $P(x^{(t+1)}, R^{(t)})$ при фиксированном $R^{(t)}$. В качестве начальной точки используемся $x^{(t)}$, а в качестве параметра окончания шага - константа ε_2.

Шаг 4. Проверить, выполняется ли условие $\left| P(x^{(t+1)}, R^{(t)}) - P(x^{(t)}, R^{(t-1)} \right| \le \varepsilon_3$. Если выполняется, положить $x^*= x^{(t+1)}$ и закончить процесс решения. В противном случае перейти к шагу 5.

Шаг 5. Положить $R^{(t+1)}= R^{(t)}+\Delta R$ в соответствии с каким–либо правилом пересчета и перейти к шагу 2.

Четких рекомендаций по выбору ΔR нет. Либо последовательно увеличивают R на некоторое число ΔR, либо в качестве R используют возрастающие степени какого–либо числа (например 10; 100; 1000 и т.д.).

При реализации шага 3 может быть использована любая процедура оптимизации: метод Хука – Дживса, поиск по симплексу. Если имеется возможность вычисления $\frac{\partial P}{\partial x_i}$, то применимы любые градиентные процедуры поиска.

2. Реализация программного комплекса

Комплекс для изучения и исследования методов условной оптимизации реализован в среде MATLAB.

После запуска комплекса на появившейся экранной форме выбирается пункт, соответствующий методам условной оптимизации, активизация которого приводит к появлению другой экранной формы.

Работа в появившемся окне аналогична работе с окнами предыдущих пунктов комплекса, снабжена соответствующей информацией и не требует дополнительных пояснений.

3. Выполнение работы

1. Ознакомиться с работой программного комплекса .
2. Найти решение предварительно подобранной тестовой задачи для всех видов штрафов.
3. Получить задание у преподавателя на выполнение работы.
4. Исследовать влияние видов штрафов и их параметров на точность получаемых решений.
5. Оформить отчёт с результатами решений и исследований методов.

Библиографический список

1. Реклейтис Г., Рейвиндран А., Рэгсдел К. Оптимизация в технике: В 2-х кн. Кн.1. Пер.с англ..-М.: Мир, 1986.-300с.
2. Банди Б. Методы оптимизации.-М.: Радио и связь, 1988.-128с.
3. Химмельблау Д. Прикладное нелинейное прграммирование.-М.: Мир, 1975.-534с.
4. Кузнецов Ю.Н.и др. Математическое программирование.-М.: Высш.шк., 1980.-300с
5. Моисеев Н.Н. и др. Методы оптимизации.-М.: Наука, 1978.-352с.
6. Дегтярёв Ю.И. Исследование операций.-М.: Высш.шк., 1986.-320с.
7. Гилл Ф. И др. Практическая оптимизация.-М.: Мир, 1985.-510с.
8. Таха Х. Введение в исследование операций.-М.:Мир, 1985.-232с.
9. Акулич И.Л. Матаматическое программирование.-М.:Высш.шк.,1986.-319.

ЛАБОРАТОРНАЯ РАБОТА №4

РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Цель работы: **изучение методов решения задач линейного программирования с последующим применением их к исследованию прикладных задач.**

1 Общая задача линейного программирования

Общая задача линейного программирования (ЛП) состоит в нахождении экстремального значения (максимума или минимума) линейной функции

$$Z = C_1 x_1 + C_2 x_2 + \ldots + C_n x_n \tag{1.1}$$

от n вещественных переменных при наложенных ограничениях:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1j}x_j + \ldots + a_{1n}x_n \le \; =, \ge) b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2j}x_j + \ldots + a_{2n}x_n \le (=, \ge) b_2 \\ \ldots \\ a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{ij}x_j + \ldots + a_{in}x_n \le (=, \ge) b_i \\ \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mj}x_j + \ldots + a_{mn}x_n \le (=, \ge) b_m \end{cases}, \tag{1.2}$$

$$x_j \ge 0 \;\; (j = 1, 2, \ldots, n \;\;, \tag{1.3}$$

где a_{ij}, b_i и C_j – заданные постоянные величины.

Линейную функцию, для которой ищется экстремальное значение, принято называть *целевой функцией*.

В системе ограничений могут одновременно встречаться знаки *меньше или равно*, *равно*, *больше или равно*.

Общая задача имеет несколько форм записи.

Векторная форма записи задачи линейного программирования имеет следующий вид:

минимизировать (максимизировать) линейную функцию

$$Z \quad \boldsymbol{CX} \tag{1.4}$$

при ограничениях

$$\boldsymbol{A}_1 x_1 \quad \boldsymbol{A}_2 x_2 + \ldots + \boldsymbol{A}_n x_n \le \; =, \;\;) \boldsymbol{A}_0, \; \boldsymbol{X} \ge 0, \tag{1.5}$$

где $\boldsymbol{C} = (c_1, \;\; c_2, \;\; \ldots, \;\; c_n \;;$

$\boldsymbol{X} = \;\; x_1, \;\; x_2, \;\; \ldots, \;\; x_n \;;$

$\boldsymbol{CX}$ – скалярное произведение.

Векторы

$$\boldsymbol{A}_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{pmatrix}, \quad \boldsymbol{A}_2 = \begin{pmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{pmatrix}, \quad \ldots, \quad \boldsymbol{A}_n = \begin{pmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{pmatrix}, \quad \boldsymbol{A}_0 = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \tag{1.6}$$

состоят соответственно из коэффициентов при неизвестных и свободных членов.

Матричная форма записи задачи линейного программирования предполагает нахождение минимального (максимального) значения линейной функции

$$Z \quad \boldsymbol{CX} \tag{1.7}$$

при ограничениях

$$\boldsymbol{AX} \le (\;\;, \ge)\boldsymbol{A}_0, \; \boldsymbol{X} \ge 0, \tag{1.8}$$

где $\boldsymbol{C} = (c_1 \quad c_2 \quad \ldots \quad c_n)$ – матрица-строка;

$$\boldsymbol{X} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \; \boldsymbol{A}_0 = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$ – матрица-столбец;

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix}$$ – матрица коэффициентов системы ограничений.

Пример 1

Найти максимальное значение линейной функции $Z = -x_1 \quad x_2 + 3x_3$ при ограничениях

$$\begin{cases} 2x_1 & -x_2 & +x_3 & \le 1 \\ 4x_1 & -2x_2 & +x_3 & \ge -2 \\ 3x_1 & & +x_3 & \le 5 \end{cases}.$$

Для данной задачи векторы имеют вид:

$$\boldsymbol{C} = (-1 \quad 2 \quad 3),$$

$$\boldsymbol{X} = (x_1 \quad x_2 \quad x_3),$$

$$\boldsymbol{A}_1 = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix}, \quad \boldsymbol{A}_2 = \begin{pmatrix} -1 \\ -2 \\ 0 \end{pmatrix}, \quad \boldsymbol{A}_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad \boldsymbol{A}_0 = \begin{pmatrix} 1 \\ -2 \\ 5 \end{pmatrix}.$$

Матрица системы ограничений выглядит следующим образом:

$$\boldsymbol{A} = \begin{pmatrix} 2 & -1 & 1 \\ 4 & -2 & 1 \\ 3 & 0 & 1 \end{pmatrix}.$$

2 Симплексный метод решения задачи линейного программирования

2.1 Стандартная форма задачи линейного программирования

Задачи ЛП, представленные различным образом, могут быть приведены к стандартной форме. Стандартная форма предполагает минимизацию целевой функции

$$Z = C_1x_1 + C_2x_2 + \dots + C_nx_n \tag{2.1}$$

при ограничениях, имеющих вид равенств:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1j}x_j + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2j}x_j + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{ij}x_j + \dots + a_{in}x_n = b_i \\ \dots \end{cases},$$

$$+a_{m2}x_2+\square+a_{mj}x_j+\square+a_{mn}x_n=b_m$$

(2.2)

$$x_j \quad 0 \quad j = 1,2,\ldots,n \quad , \tag{2.3}$$

$$b_i \ge 0 \quad (i = 1,2,\ldots,m \quad . \tag{2.4}$$

Если для задачи в форме (2.1)-(2.4) выполняется условие $m = n$, то решение задачи ЛП сводится к решению системы уравнений (2.2). При этом задача не будет иметь решений, если условие (2.3) не выполняется или система уравнений не имеет решения. Задача ЛП не будет иметь решения и в случае, когда $\quad > n$, так как при этом система ограничений не имеет решения.

Далее рассматриваются задачи, для которых выполняется условие $m < n$.

Чтобы от задачи нахождения максимального значения целевой функции (1.1) перейти к задаче минимизации, достаточно взять все коэффициенты C_j целевой функции с обратными знаками. Для обратного перехода после нахождения минимума результат так же необходимо инвертировать.

Для перехода от ограничения типа *меньше или равно* к равенству в него необходимо ввести дополнительную неотрицательную переменную со знаком «плюс»:

$$a_{i1}x_1 + \ldots + a_{ij}x_j + \ldots + a_{in}x_n \le b_1 \quad \Rightarrow \quad a_{i1}x_i + \qquad a_{ij}x_j + \ldots + a_{in}x_n + x_{n+1} = b_i. \quad \mathbf{(2.5)}$$

Для перехода от ограничения типа *больше или равно* к равенству в него необходимо ввести дополнительную неотрицательную переменную со знаком «минус»:

$$a_{i1}x_1 + \ldots + a_{ij}x_j + \ldots + a_{in}x_n \ge b_1 \quad \Rightarrow \quad a_{i1}x_i + \qquad a_{ij}x_j + \ldots + a_{in}x_n - x_{n+1} = b_i. \quad \mathbf{(2.6)}$$

При этом в каждое неравенство вводится своя $(n \quad i$ -я дополнительная переменная.

Все равенства, имеющие отрицательные свободные члены, делятся на -1, для того чтобы выполнялось условие (2.4).

Пример 2

Преобразуем задачу, приведенную в примере 1 к стандартной форме. Система ограничений выглядит следующим образом:

$$\begin{cases} 2x_1 & -x_2 & +x_3 & \le 1 \\ 4x_1 & -2x_2 & +x_3 & \ge -2 \\ 3x_1 & & +x_3 & \le 5 \end{cases}.$$

Умножим второе ограничение на -1, для обеспечения условия неотрицательности свободных членов в системе ограничений:

$$\begin{cases} 2x_1 & -x_2 & +x_3 & \le 1 \\ -4x_1 & +2x_2 & -x_3 & \le 2 \\ 3x_1 & & +x_3 & \le 5 \end{cases}.$$

Введем в первое неравенство дополнительную переменную $x_4 \ge 0$ со знаком «плюс», во второе - $x_5 \quad 0$ и в третье - $x_6 \ge 0$ также со знаком «плюс». В результате получим систему ограничений задачи в стандартной форме:

$$\begin{cases} 2x_1 & -x_2 & +x_3 & +x_4 & & & =1 \\ -4x_1 & +2x_2 & -x_3 & & +x_5 & & =2 \\ 3x_1 & & +x_3 & & & +x_6 & =5 \end{cases}.$$

При этих ограничениях нужно найти минимальное значение функции с обратными коэффициентами:

$$Z = x_1 - x_2 - 3x_3 .$$

2.2 Понятие плана задачи линейного программирования

Для задачи ЛП, представленной в стандартной форме, введено понятие *плана*.

***Планом,* или *допустимым решением* задачи линейного программирования, называется вектор $X = \; x_1, \;\; x_2, \qquad , \;\; x_n)$, удовлетворяющий условиям (2.2), (2.3) и (2.4).**

План называется *опорным,* если векторы A_i $(i = 1,2,\square,m)$, входящие в разложение вида (1.5) с положительными коэффициентами x_i (для стандартной формы), являются линейно независимыми. Число положительных компонент опорного плана не может превышать m. Опорный план называется невырожденным, если он содержит m положительных компонент, в противном случае опорный план называется *вырожденным*.

***Оптимальным* планом или оптимальным решением задачи линейного программирования называется план, доставляющий наименьшее значение линейной функции (2.1).**

Множество всех планов задачи линейного программирования (если они существуют) является выпуклым многогранником. Каждой угловой точке многогранника решений соответствует опорный план. Каждый опорный

план определяется системой m линейно независимых векторов, содержащихся в данной системе из n векторов $A_1, A_2, \ldots, A_n$. Существует такая угловая точка многогранника решений, в которой линейная функция достигает своего наименьшего значения.

Для отыскания оптимального плана необходимо исследовать только опорные планы. Верхняя граница количества опорных планов, содержащихся в данной задаче, определяется числом сочетаний C_n^m. При больших m и n найти оптимальный план, перебирая все опорные планы задачи, очень трудно.

Симплексный метод предоставляет схему, позволяющую осуществлять упорядоченный переход от одного опорного плана к другому. Этот метод позволяет, исходя из известного опорного плана задачи, за конечное число шагов получить ее оптимальный план. Каждый из шагов (или итераций) состоит в нахождении нового плана, которому соответствует меньшее значение линейной функции, чем значение этой же функции в предыдущем плане. Если задача не обладает планами или ее линейная функция не ограничена на многограннике решений, то симплексный метод позволяет установить это в процессе решения.

2.3 Построение опорных планов

Для построения первоначального опорного плана необходимо выделить в системе ограничений (2.2) m линейно независимых векторов. Наиболее просто это произвести, преобразовав систему ограничений таким образом, чтобы в ней появилось m единичных векторов (единичная подматрица):

$$\begin{cases} x_1 \qquad\qquad\qquad + a_{1,m+1}x_{m+1} + \ldots + a_{1n}x_n = b_1 \\ \qquad x_2 \qquad\qquad + a_{2,m+1}x_{m+1} + \ldots + a_{2n}x_n = b_2 \\ \qquad\qquad \ldots \\ \qquad\qquad\qquad x_m + a_{m,m+1}x_{m+1} + \ldots + a_{mn}x_n = b_n \end{cases}, \quad b_i \geq 0 \quad (j = 1, 2, \ldots, m. \qquad (2.7)$$

В векторной форме система (2.7) имеет следующий вид:

$$x_1\boldsymbol{A}_1 \quad x_2\boldsymbol{A}_2 + \ldots + x_m\boldsymbol{A}_m + x_{m+1}\boldsymbol{A}_{m+1} \quad \ldots + x_n\boldsymbol{A}_n \quad \boldsymbol{A}_0, \qquad (2.8)$$

где

$$\boldsymbol{A}_1 = \begin{pmatrix} 1 \\ 0 \\ \ldots \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{A}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \ldots \\ 0 \end{pmatrix}, \quad \ldots, \quad \boldsymbol{A}_m = \begin{pmatrix} 0 \\ 0 \\ \ldots \\ 0 \\ 1 \end{pmatrix};$$

$$\boldsymbol{A}_{m+1} = \begin{pmatrix} a_{1,m+1} \\ a_{2,m+1} \\ \vdots \\ a_{m,m+1} \end{pmatrix}, \quad \ldots, \quad \boldsymbol{A}_n = \begin{pmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{pmatrix}, \quad \boldsymbol{A}_0 = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}.$$

Векторы $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_m$ – линейно независимые единичные вектора m-мерного пространства. Они образуют базис этого пространства. Любой из векторов $\boldsymbol{A}_j$ можно представить как линейную комбинацию базисных векторов, причем единственным образом:

$$\boldsymbol{A}_j = \sum_{i=1}^{m} x_{ij} \boldsymbol{A}_i, \; i = 0, 1, 2, \ldots n. \tag{2.9}$$

Если в разложении (2.7) за базисные неизвестные выбираем $x_1, x_2, \ldots, x_m$, свободные неизвестные $x_{m+1}, \ldots, x_n$ приравниваем нулю и, учитывая, что $b_i \geq 0$ $(j = 1, 2, \ldots, m)$, а векторы $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_m$ – единичные, получаем первоначальный опорный план:

$$\boldsymbol{X}_0 = (x_1 = b_1; \;\; x_2 = b_2; \;\; \ldots \;\; x_m = b_m; \;\; x_{m+1} = 0; \;\; \ldots \;\; x_n = 0). \tag{2.10}$$

Если задаться некоторой величиной $\theta > 0$, то вектор

$$\boldsymbol{X}_1 = (x_1 - \theta x_{1,m+1}; \;\; x_2 - \theta x_{2,m+1}; \;\; \ldots \;\; x_m - \theta x_{m,m+1}; \;\; \theta; \;\; 0 \;\; \ldots \;\; 0)$$

также является планом, если его компоненты неотрицательны.

Так как $\theta > 0$, то все компоненты вектора $\boldsymbol{X}_1$, в которые входят неположительные $x_{i,m+1}$, неотрицательны. Необходимо определить такое $\theta > 0$, при котором для всех $x_{i,m+1} > 0$ выполняется условие

$$x_i - \theta x_{i,m+1} \geq 0. \tag{2.11}$$

Из (2.11) следует $\theta \leq x_i / x_{i,m+1}$, следовательно, вектор $\boldsymbol{X}_1$ – план задачи для любого θ, удовлетворяющего условию

$$0 < \theta \leq \min\left[x_i / x_{i,m+1}\right], \tag{2.12}$$

где минимум берется по i, для которых $x_{i,m+1} > 0$.

Так как опорный план не может содержать $m+1$ положительных компонент, поэтому в плане $\boldsymbol{X}_1$ необходимо обратить в нуль по крайней мере одну из компонент. Если положить

$$= \quad _0 \quad min\ x_i / x_{i,m+1} \ , \qquad (2.13)$$

то компонента плана $\boldsymbol{X}_1$, для которой достигается минимум обращается в нуль, тем самым осуществляется переход к новому опорному плану:

$$\boldsymbol{X}_1 = 0; \quad x_2`; \quad \square \quad x_m`; \quad x_{m+1}`; \quad 0 \quad \square \quad 0 \ . \qquad (2.14)$$

2.4 Отыскание оптимального плана

Если задача линейного программирования (2.1)-(2.4) обладает планами и каждый ее опорный план невырожден, тогда для опорного плана (2.10) выполняется следующее соотношение:

$$x_1\boldsymbol{A}_1 \quad x_2\boldsymbol{A}_2 + \square \quad x_m\boldsymbol{A}_m = \boldsymbol{A}_0 , \qquad (2.15)$$

$$x_1C_1 + x_2C_2 + \square + x_mC_m = Z\big(\boldsymbol{A}_0 \ , \qquad (2.16)$$

где все $x_i \quad 0$;

$Z\big(\boldsymbol{A}_0$ – значение целевой функции, соответствующее этому плану.

Разложение любого вектора $\boldsymbol{A}_j$ по векторам данного базиса единственное:

$$x_{1j}\boldsymbol{A}_1 + x_{2j}\boldsymbol{A}_2 + \square + x_{mj}\boldsymbol{A}_m = \boldsymbol{A}_j \ \big(j = 1,2,\square,n \ , \qquad (2.17)$$

этому разложению соответствует и единственное значение линейной функции

$$x_{1j}C_1 \quad x_{2j}C_2 \quad \square + x_{mj}C_m \quad Z_j \ \big(j = 1,2,\square,n \ , \qquad (2.18)$$

где Z_j – значение линейной функции, если в нее вместо неизвестных подставить соответствующие коэффициенты разложения j-го вектора по векторам базиса.

Если для некоторого плана $\tilde{\boldsymbol{X}}$ разложения всех векторов $\boldsymbol{A}_j \quad j \quad 1,2, \quad ,n\big)$ в данном базисе удовлетворяет условию

$$Z_j \quad C_j \le 0 , \qquad (2.19)$$

то план является оптимальным.

Неравенства (2.19) являются условием оптимальности плана задачи, решаемой на отыскание минимального значения линейной функции, а значения $Z_j \quad C_j$ называются *оценками* плана.

Таким образом, для того чтобы план задачи был оптимальным, необходимо и достаточно, чтобы его оценки были неположительными.

3 Алгоритм симплексного метода

3.1 Построение симплексной таблицы

Следующим шагом после того, как задача приведена к стандартной форме, в системе ограничений выделена единичная подматрица и получен первоначальный опорный план в виде (2.10), необходимо исследовать этот план на оптимальность. Для этого необходимо векторы $\boldsymbol{A}_j$ $\left(j \quad 1,2,\ldots,n\right)$ системы разложить по векторам базиса $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_m$ и подсчитать значения оценок $Z_j \quad C_j$. Базис является единичным, поэтому коэффициентами разложения вектора $\boldsymbol{A}_j$ по базису служат его компоненты, т.е. $x_{ij} = a_{ij}$ $i = 1,2,\ldots,m;\ j = 1,2,\ldots,n$. Это означает, что в столбцы $\boldsymbol{A}_j$ $\left(j \quad 1,2,\ldots,n\right)$ таблицы нужно записать матрицу ограничений вида (2.7). Дальнейшие вычисления удобнее проводить, если условия задачи и первоначальные данные, полученные после определения первого опорного плана, записать в симплексную таблицу (табл.1).

Таблица 1

I	Базис	**C**	$\boldsymbol{A}_0$	C_1	C_2	…	C_l	…	C_m	C_{m+1}	…	C_j	…	C_k	…	C_n
		Базиса		$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	…	$\boldsymbol{A}_l$	…	$\boldsymbol{A}_m$	$\boldsymbol{A}_{m+1}$	…	$\boldsymbol{A}_j$	…	$\boldsymbol{A}_k$	…	$\boldsymbol{A}_n$
1	$\boldsymbol{A}_1$	C_1	X_1	*1*	*0*	…	*0*	…	*0*	$x_{1,m+1}$	…	x_{1j}	…	x_{1k}	…	x_{1n}
2	$\boldsymbol{A}_2$	C_2	X_2	*0*	*1*	…	*0*	…	*0*	$x_{2,m+1}$	…	x_{2j}	…	x_{2k}	…	x_{2n}
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
l	$\boldsymbol{A}_l$	C_l	x_l	*0*	*0*	…	*1*	…	*0*	$x_{l,m+1}$	…	x_{lj}	…	$\boldsymbol{x_{lk}}$	…	x_{ln}
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
m	$\boldsymbol{A}_m$	C_m	x_m	*0*	*0*	…	*0*	…	*1*	$x_{m,m+1}$	…	x_{mj}	…	x_{mk}	…	x_{mn}
M+1	Z_j-C_j		Z_0	*0*	*0*	…	*0*	…	*0*	Z_{m+1}-C_{m+1}	…	Z_j-C_j	…	Z_k-C_k	…	Z_n-C_n

В столбце **C** *базиса* записываются коэффициенты линейной функции, соответствующие векторам базиса. В столбце $\boldsymbol{A}_0$ – первоначальный опорный план $\boldsymbol{X}_0$, в нем же в результате вычислений получается оптимальный план, а столбцах $\boldsymbol{A}_j$ $\left(j = 1,2,\ldots,n\right.$ записываются коэффициенты разложения j-го вектора по базису, обозначаемые в дальнейшем через $\boldsymbol{X}_j$

В $(m+1)$-й строке в столбце $\boldsymbol{A}_0$ записываются значения линейной функции $Z(\boldsymbol{X}_0$, которое она принимает при найденном опорном плане, а в столбцах $\boldsymbol{A}_j$ – значения оценок $Z_j - C_j$.

Функции $Z \ \boldsymbol{X}_0)$ и $Z_j \quad Z(\boldsymbol{X}_j)$ находим, подставляя в линейную функцию соответственно компоненты опорного плана и коэффициенты разложения j-го вектора по векторам базиса, поэтому эти значения в табл.1 можно получить как скалярное произведение:

$$Z(\boldsymbol{X}_0) = \boldsymbol{C}_б \boldsymbol{X}_0 = \sum_{i=1}^{m} C_i x_i , \tag{3.1}$$

$$Z_j = \boldsymbol{C}_б \boldsymbol{X}_j = \sum_{i=1}^{m} C_i x_{ij} , \ j = 1,2,\square,n , \tag{3.2}$$

где C_j – коэффициенты линейной функции, соответствующие векторам базиса.

Пример 3.1

Составим первоначальную симплексную таблицу для задачи в примере 2.

Единичные векторы $\boldsymbol{A}_4$, $\boldsymbol{A}_5$ и $\boldsymbol{A}_6$ образуют единичную подматрицу и составляют базис первоначального плана, свободные неизвестные приравниваются нулю. В результате получается первоначальный опорный план:

$$\boldsymbol{X}_0^{(1)} = (0; \quad 0; \quad 0; \quad 1; \quad 2; \quad 5).$$

Вычислим значения $(m+1)$-й строки:

$$Z(\boldsymbol{X}_0 \quad \boldsymbol{C}_б \boldsymbol{X}_0 \quad 0;$$

$$Z_1 \quad \boldsymbol{C}_б \boldsymbol{X}_1 = 0; \quad Z_2 \quad \boldsymbol{C}_б \boldsymbol{X}_2 \quad 0; \quad Z_3 = \boldsymbol{C}_б \boldsymbol{X}_3 = 0;$$

$$Z_1 - C_1 = 0 \quad 1 \quad 1; \quad Z_2 - C_2 = 0 + 1 \quad 1; \quad Z_3 \quad C_3 = 0 + 3 \quad 3 .$$

Таблица 2

i	Базис	**C** базиса	$\boldsymbol{A}_0$	$C_1=1$	$C_2=-1$	$C_3=-3$	$C_4=0$	$C_5=0$	$C_6=0$
				$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	$\boldsymbol{A}_5$	$\boldsymbol{A}_6$
1	$\boldsymbol{A}_4$	0	1	2	-1	**1**	1	0	0
2	$\boldsymbol{A}_5$	0	2	-4	2	-1	0	1	0
3	$\boldsymbol{A}_6$	0	5	3	0	1	0	0	1
$m+1$	Z_j-C_j		0	-1	1	3	0	0	0

3.2 Поиск разрешающего элемента симплексной таблицы

После составления табл.1 анализируется $(m+1)$-я строка. Если для всех $j=1,2,\ldots,n$ значения оценок $Z_j - C_j$ меньше нуля, то опорный план $\boldsymbol{X}_0$ оптимальный и минимальное значение линейной функции равно $Z(\boldsymbol{X}_0)$. В противном случае можно, включив в базис вектор, соответствующий положительной оценке, построить другой опорный план, которому соответствует меньшее значение линейной функции.

Если положительных оценок несколько, то в базис должен быть включен вектор, которому соответствует $max\,(\theta_{0j}(Z_j - C_j))$, так как уменьшение значения целевой функции происходит именно на величину $\theta_{0j}(Z_j - C_j)$. Максимум берется по тем j, для которых оценка положительна и θ_{0j} определяется для каждого j. Это дает возможность на данном шаге перейти к вершине многогранника решений, связанной с наибольшим уменьшением линейной функции.

Пусть в базис включается k-й вектор, тогда исключается из базиса тот вектор, которому соответствует $\theta_{0k} = min[x_i / x_{ik}]$ $(x_{ik} > 0)$. Пусть это условие выполняется для вектора базиса $\boldsymbol{A}_l$. Элемент x_{lk} называется *разрешающим*, а столбец и строка, на пересечении которых он находится, – *направляющими*. Новому опорному плану соответствует базис, состоящий из векторов $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_{l-1}, \boldsymbol{A}_k, \boldsymbol{A}_{l+1}, \ldots, \boldsymbol{A}_m$.

3.3 Переход к новому опорному плану

Чтобы вычислить новый опорный план и проверить его на оптимальность, необходимо все векторы разложить по векторам нового базиса. Новый опорный план и разложения векторов в новом базисе при $j = 1, 2, \ldots, n$ определяется по формулам:

$$\begin{cases} x'_{ij} = x_{ij} - \dfrac{x_{ij}}{x_{lk}} & (i \neq l) \\ x'_{ij} = \dfrac{x_{ij}}{x_{lk}} & (i = j) \end{cases}, \tag{3.3}$$

которые являются формулами полных исключений Жордана-Гаусса.

Таким образом, чтобы получить коэффициенты разложения векторов $\boldsymbol{A}_0$, $\boldsymbol{A}_j$ $(j = 1,2,\ldots,n)$ по векторам нового базиса, значения оценок нового опорного плана и значения линейной функции, нужно разделить все элементы направляющей строки на разрешающий элемент и, производя одно полное преобразование по методу Жордана-Гаусса, с помощью этой преобразованной строки составить новую симплексную таблицу.

3.4 Условия окончания преобразований

Если в симплексной таблице в $m+1$ -й строке все оценки $\left(Z_j - C_j\right) \leq 0$, то полученный план является оптимальным; если же имеются положительные оценки, то ищется следующий опорный план.

Если хотя бы для одной положительной оценки коэффициенты разложения x_{ij} соответствующего вектора неположительные, то линейная функция не ограничена на многограннике решений. При этом линейная функция может принимать сколь угодно малое значение.

Процесс продолжается либо до получения оптимального плана, либо до установления неограниченности линейной функции решаемой задачи. Если среди оценок оптимального плана нулевые только оценки, соответствующие базисным векторам, то это говорит о единственности оптимального плана.

Если система ограничений задачи линейного программирования содержит единичный базис, то для решения задачи необходимо примерно итераций.

Пример 3.2

Продолжим рассмотрение задачи из примера 3.1.

В $m+1$ -й строке первоначальной таблицы (табл.2) имеются две положительные оценки. Они соответствуют векторам $\boldsymbol{A}_2$ и $\boldsymbol{A}_3$. Это означает, что первоначальный план не является оптимальным и его можно улучшить, включив в базис вектор с $max\left[\theta_{0j}\left(Z_j \quad C_j \;> 0\right]\right.$. Среди коэффициентов разложения векторов $\boldsymbol{A}_2$ и $\boldsymbol{A}_3$ по базису имеются положительные, поэтому $\theta_{02} > 0$ и $_{03} > 0$, которые исключают из базиса хотя бы один из векторов, существуют. Найдем эти значения:

$$\theta_{02} = 2/2;$$

$$\theta_{03} = min\ 1/1, \quad 5/1 \quad\; 1;$$

$$\theta_{02}\ Z_2 - C_2) = 1\ \ 1 = 1;$$

$$\theta_{03}(Z_3 - C_3 \quad\; 1 \cdot 3 \quad 3;$$

$$max\ 1, \quad 3 \quad\; 3.$$

Следовательно, разрешающим элементом служит число 1, стоящее на пересечении первой строки и третьего столбца. Это означает, что необходимо вектор $\boldsymbol{A}_3$ включить в базис, а вектор $\boldsymbol{A}_4$ исключить.

Составим вторую симплексную таблицу (табл.3).

Таблица 3

i	Базис	**C** базиса	$\boldsymbol{A_0}$	$C_1=1$	$C_2=-1$	$C_3=-3$	$C_4=0$	$C_5=0$	$C_6=0$
				$\boldsymbol{A_1}$	$\boldsymbol{A_2}$	$\boldsymbol{A_3}$	$\boldsymbol{A_4}$	$\boldsymbol{A_5}$	$\boldsymbol{A_6}$
1	$\boldsymbol{A_3}$	*-3*	*1*	*2*	*-1*	*1*	*1*	*0*	*0*
2	$\boldsymbol{A_5}$	*0*	*3*	*-2*	$\boxed{\mathbf{1}}$	*0*	*1*	*1*	*0*
3	$\boldsymbol{A_6}$	*0*	*4*	*1*	*1*	*0*	*-1*	*0*	*1*
m+1	Z_j-C_j		*-3*	*-7*	*4*	*0*	*-3*	*0*	*0*

Подсчитаем новые элементы направляющей строки. Для этого старые элементы направляющей строки разделим на разрешающий элемент (на *1*) и с помощью полученной строки произведем одно преобразование по методу полных исключений, т.е. прибавим ко второй и вычтем из третьей.

Значения в $(m+1)$-й строке рассчитываются аналогично примеру 3.1.

В табл.3 получен второй опорный план $\boldsymbol{X}_0^{(2)} = (0;\ 0;\ 1;\ 0;\ 3;\ 4)$, которому соответствует значение линейной функции $Z\left(\boldsymbol{X}_0^{(2)}\right) = -3$. Второй план не является оптимальным, так как оценка в $(m+1)$-й строке $(Z_2 \quad C_2 \ = 4 \quad 0$.

Определяем $\theta_{02} = min\ 3/1,\ \ 4/1 \quad 3$. Число *1*, стоящее на пересечении второго столбца и второй строки, является разрешающим элементом, вектор $\boldsymbol{A}_5$ исключается из базиса. Составляем третью симплексную таблицу (табл.4).

Таблица 4

I	Базис	**C** базиса	$\boldsymbol{A_0}$	$C_1=1$	$C_2=-1$	$C_3=-3$	$C_4=0$	$C_5=0$	$C_6=0$
				$\boldsymbol{A_1}$	$\boldsymbol{A_2}$	$\boldsymbol{A_3}$	$\boldsymbol{A_4}$	$\boldsymbol{A_5}$	$\boldsymbol{A_6}$
1	$\boldsymbol{A_3}$	*-3*	*4*	*0*	*0*	*1*	*2*	*1*	*0*
2	$\boldsymbol{A_2}$	*-1*	*3*	*-2*	*1*	*0*	*1*	*1*	*0*
3	$\boldsymbol{A_6}$	*0*	*1*	$\boxed{\mathbf{3}}$	*0*	*0*	*-2*	*-1*	*1*
m+1	Z_j-C_j		*-15*	*1*	*0*	*0*	*-7*	*-4*	*0*

В табл.4 получен третий опорный план $\boldsymbol{X}_0^{(3)} = (0 \quad 3;\ \ 4;\ \ 0;\ \ 0;\ \ 1$, которому соответствует значение линейной функции $Z\ \boldsymbol{X}_0^{(3)} \ = -15$. Третий план так же не является оптимальным, так как $(Z_1 - C_1) = 1 > 0$. Составляется следующая симплексная таблица (табл.5).

Таблица 5

i	Базис	**C** базиса	$\boldsymbol{A_0}$	$C_1=1$	$C_2=-1$	$C_3=-3$	$C_4=0$	$C_5=0$	$C_6=0$
				$\boldsymbol{A_1}$	$\boldsymbol{A_2}$	$\boldsymbol{A_3}$	$\boldsymbol{A_4}$	$\boldsymbol{A_5}$	$\boldsymbol{A_6}$
1	$\boldsymbol{A_3}$	*-3*	*4*	*0*	*0*	*1*	*2*	*1*	*0*
2	$\boldsymbol{A_2}$	*-1*	$11/3$	*0*	*1*	*0*	$-1/3$	$1/3$	$2/3$
3	$\boldsymbol{A_1}$	*1*	$1/3$	*1*	*0*	*0*	$-1/2$	$-1/3$	$1/3$

$m+1$	Z_j-C_j	$-46/3$	0	0	0	$-19/3$	$-11/3$	$1/3$

По табл.5 можно сделать вывод, что план

$$X_0^{(4} = 1/3; \quad 11/3; \quad 4; \quad 0; \quad 0; \quad 0$$

является оптимальным и притом единственным. Минимальное значение линейной функции в находится в точке $X_0^{4)}$ и равно $-46/3$.

4. Двойственные задачи линейного программирования

4.1. Прямая и двойственная задача

С каждой задачей линейного программирования тесно связана другая линейная задача, называемая двойственной или сопряженной по отношению к исходной или прямой /1, 5/. Дадим определение двойственной задачи по отношению к исходной задаче линейного программирования, состоящей в нахождении максимального значения функции

$$Z = c_1 x_1 + c_2 x_2 + \quad + c_n x_n \tag{4.1}$$

при ограничениях

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \square + a_{1n}x_n \le b_1 \\ a_{21}x_1 + a_{22}x_2 + \square + a_{2n}x_n \le b_2 \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \square + a_{mn}x_n \le b_m \end{cases} \tag{4.2}$$

$$(j = 1,2,\square,n). \tag{4.3}$$

Задача, состоящая в нахождении минимального значения функции

$$Z' = b_1 x_1 + b_2 x_2 + \square + b_m x_m \tag{4.4}$$

при ограничениях

$$\begin{cases} a_{11}y_1 + a_{21}y_2 + \square + a_{m1}y_m \le c_1 \\ a_{12}y_1 + a_{22}y_2 + \square + a_{m2}y_m \le c_2 \\ \cdots \\ a_{1n}y_1 + a_{2n}y_2 + \square + a_{mn}y_m \le c_m \end{cases}, \tag{4.5}$$

$$y_j \ge 0 \quad (i = 1,2,\square,m). \tag{4.6}$$

называется двойственной по отношению к задаче (4.1) – (4.3).

Задачи (4.1) – (4.3) и (4.4) – (4.6) образуют пару задач, называемую в линейном программировании двойственной парой.

Из сравнения двух сформулированных задач видно, что двойственная задача по отношению к исходной составляется согласно следующим правилам:

1.Целевая функция имеет противоположный тип экстремума.

2.Матрица коэффициентов двойственной задачи получается в результате транспонирования матрицы коэффициентов исходной задачи.

3.Число переменных в двойственной задаче (4.4) – (4.6) равно числу соотношений в системе (4.2) исходной задачи, а число ограничений в системе (4.5) – числу переменных в исходной задаче.

4.Коэффициентами при неизвестных в целевой функции (4.4) двойственной задачи являются свободные члены в системе (4.2), а правыми частями в системе (4.5) двойственной задачи – коэффициенты при неизвестных в целевой функции (4.1).

Многие задачи линейного программирования первоначально составляются в виде исходных или двойственных задач, поэтому имеет смысл говорить о паре двойственных задач линейного программирования.

Между оптимальными планами пары двойственных задач существует связь, которую устанавливает теорема двойственности [5]:

Если из пары двойственных задач одна обладает оптимальным планом, то и другая имеет решение, причем для экстремальных значений линейных функций выполняется соотношение

$$\min Z = \max Z' .$$

Если линейная функция одной из задач не ограничена, то другая не имеет решения.

4.2 Модели двойственных задач

Различают несимметричные и симметричные двойственные задачи. В несимметричных двойственных задачах система ограничений исходной задачи задается в виде равенств, а двойственной – в виде неравенств, причем в последней переменные могут быть и отрицательными. В симметричных задачах система ограничений как исходной так и двойственной задач задается неравенствами, причем на двойственные переменные налагается условие неотрицательности.

Математические модели пары двойственных задач могут иметь один из следующих видов.

Несимметричные задачи

Исходная задача

1. $Z_{\min} = CX;$

$AX = B;$

$X \quad 0.$

Двойственная задача

$Z'_{\max} = YB;$

$YA \quad C.$

2. $Z_{max} = CX$; $Z_{min} = YB$;

$AX \quad B$; $YA \quad C$.

$X \geq 0$.

Симметричные задачи

Исходная задача   Двойственная задача

3. $Z_{min} = CX$; $Z'_{max} = YB$;

$AX \geq B$; $YA \leq C$;

$X \geq 0$. 0.

4. $Z_{max} = CX$; $Z'_{min} = YB$;

$AX \leq B$; $YA \geq C$;

$X \geq 0$. ≥ 0.

Здесь

$X = (x_1, x_2, ..., x_n)^T$ -матрица-столбец;

$C \quad (c_1, c_2, ..., c_n)$-матрица-строка;

$B = (b_1, b_2, ..., b_m)^T$-матрица-столбец;

$Y \quad (y_1, y_2, ..., y_m)$-матрица-строка;

$A \quad (a_{ij})_{mxn}$-матрица коэффициентов системы ограничений.

В заключение приведём следующую теорему, которая используется при оценке результатов решения двойственных задач линейного программирования.

Теорема. *Если при подстановке компонент оптимального плана в систему ограничений исходной задачи i-е ограничение обращается в неравенство, то i-я компонента оптимального плана двойственной задачи равна нулю.*

Если i-я компонента оптимального плана двойственной задачи положительна, то i-е ограничение исходной задачи удовлетворяется её оптимальным решением как строгое равенство.

5. Описание лабораторного комплекса

Комплекс представляет собой программный модуль, написанный в среде Borland C++ Builder 3. Комплекс предназначен для решения задач линейного программирования и проведения лабораторной работы по изучению симплексного метода решения задачи.

Программная форма состоит из четырех страниц: «Модель», «Таблица», «Результат» и «Задача».

5.1 Страница «Модель»

Страница «Модель» предназначена для ввода модели задачи в исходном виде. Приведение к каноническому виду не требуется. Процедура преобразования задачи реализована программно. На странице расположены четыре области для ввода и две кнопки (рис. 1).

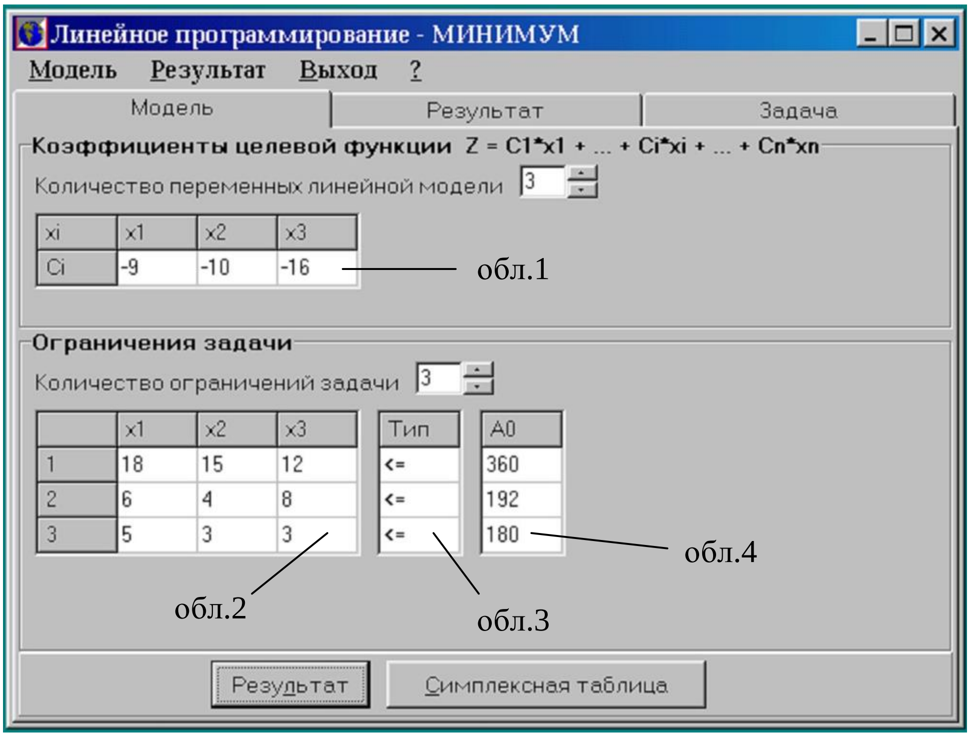

Рис. 1. Страница «Модель»

В обл.1 вводятся значения коэффициентов C_i целевой линейной функции (матрица-строка $\boldsymbol{C}$).

Обл.2 представляет собой матрицу $\boldsymbol{A}$ ограничений задачи.

В обл.3 задается тип ограничения для каждой строки матрицы $\boldsymbol{A}$. Значение каждого поля данной области могут принимать значения «==» для ограничений типа *равно*, «>=» – для ограничений типа *больше-или-равно* и «<=» – для типа *меньше-или-равно* в любом порядке. Тип можно задавать с клавиатуры либо циклически менять значения двойным нажатием левой кнопки мыши.

Обл.4 – вектор $\boldsymbol{A_0}$ свободных членов системы ограничений задачи.

Все числовые данные вводятся в формате с плавающей точкой. Точность вычислений 10^{-6}. Запись дробями не допускается. Корректность ввода проверяется автоматически.

Количество переменных модели n можно изменять в пределах от 2 до 100, количество ограничений (m) – от 1 до 100. Введенную модель можно сохранить на диск в определенном формате и затем использовать модели, вводя их из файла. Эти операции выполняются с помощью меню.

По нажатию кнопки «Результат» выводится результаты решения поставленной задачи, при условии правильной ее постановки (переход на страницу «Результат»).

По нажатию кнопки «Симплексная таблица» задача преобразуется к каноническому виду, определяется первоначальный опорный план и выводится симплексная таблица (переход на страницу «Таблица»).

5.2 Страница «Таблица»

На этой странице (рис. 2) расположена симплексная таблица, краткий комментарий и три кнопки.

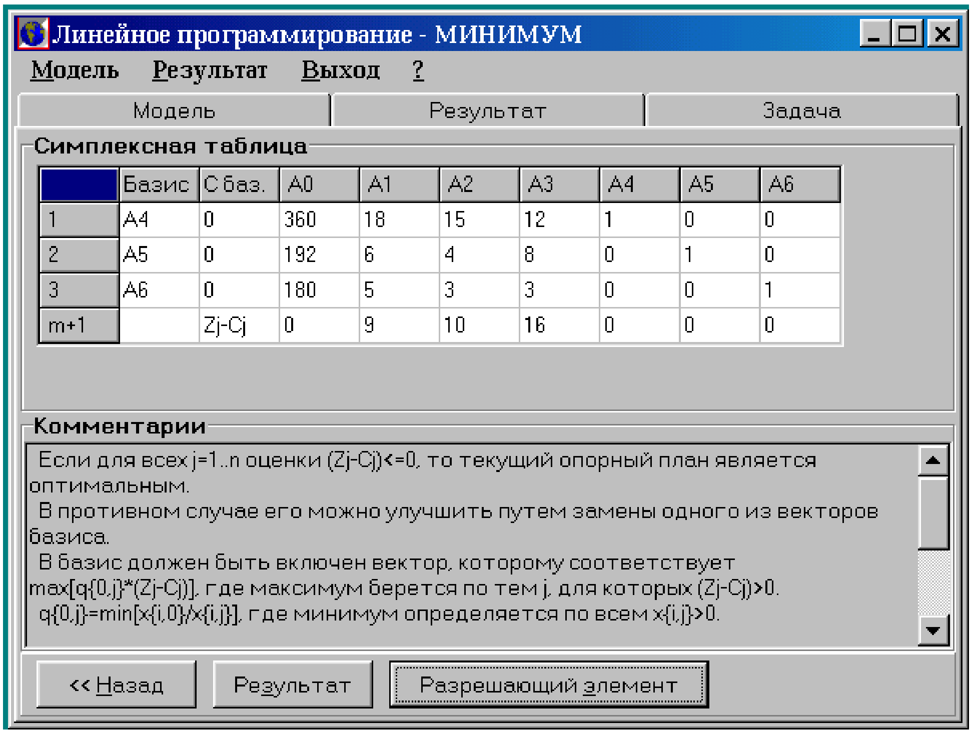

Рис. 2. Страница «Таблица»

На этой странице отображается пошаговое решение задачи. Каждый шаг состоит из определения разрешающего элемента и пересчета таблицы. Для этого на странице имеются соответствующие кнопки.

По нажатию кнопки «Разрешающий элемент» в текущей таблице определяется и подсвечивается разрешающий элемент.

По нажатию кнопки «Пересчет таблицы» осуществляется пересчет симплексной таблицы в соответствии с разрешающим элементом. Таблица содержит округленные значения всех величин, что не влияет на точность конечного результата. На последнем шаге решения выводится сообщение о результате решения задачи.

Кнопка «Назад» возвращает к предыдущей странице. Для вычисления конечного результата на любом шаге можно перейти к странице «Результат».

Кнопки «Результат» дублирует соответствующую кнопку на предыдущей странице.

5.3 Страница «Результат»

Страница «Результаты» (рис. 3) содержит текстовую формулировку обобщенной задачи линейного программирования в соответствии с моделью, заданной на странице «Модель», и результаты решения:

- минимальное значение линейной функции, если оно существует;
- оптимальные значения переменных.

Так же как и модель, результат решения можно сохранить в текстовом виде на диск и прочитать с диска сохраненные результаты с помощью меню.

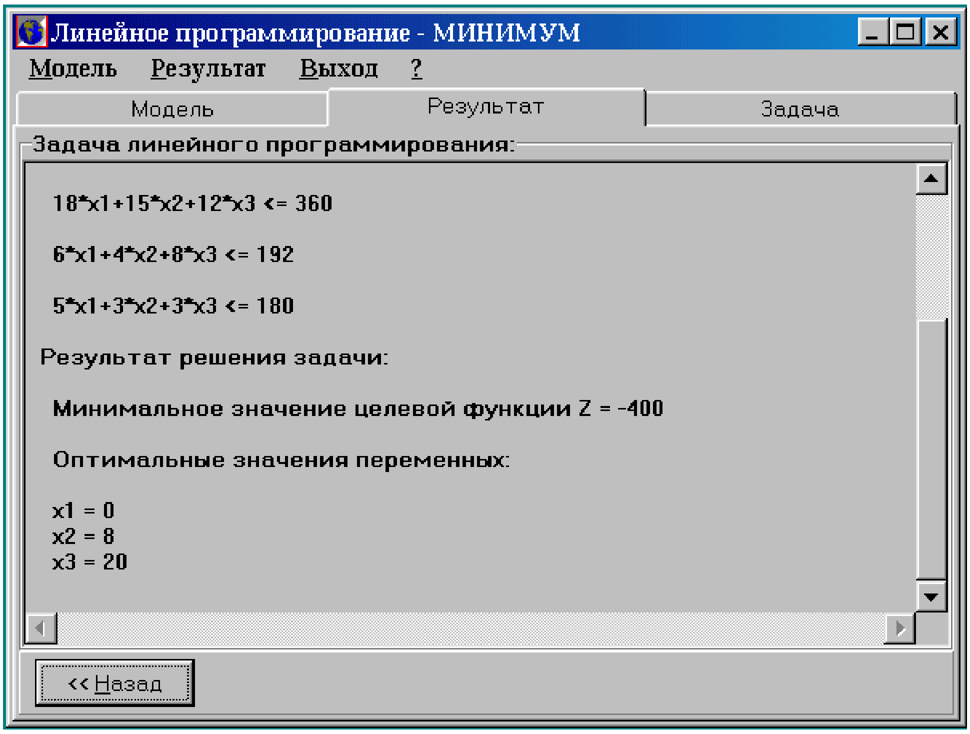

Рис. 3. Страница «Результат»

5.4 Страница «Задача»

Данная страница (рисунок 4) используется только при выполнении лабораторной работы. На странице «Задача» можно просмотреть текст задачи, в соответствии с заданным вариантом. В библиотеку занесены задачи трех уровней, по 10 вариантов для каждого.

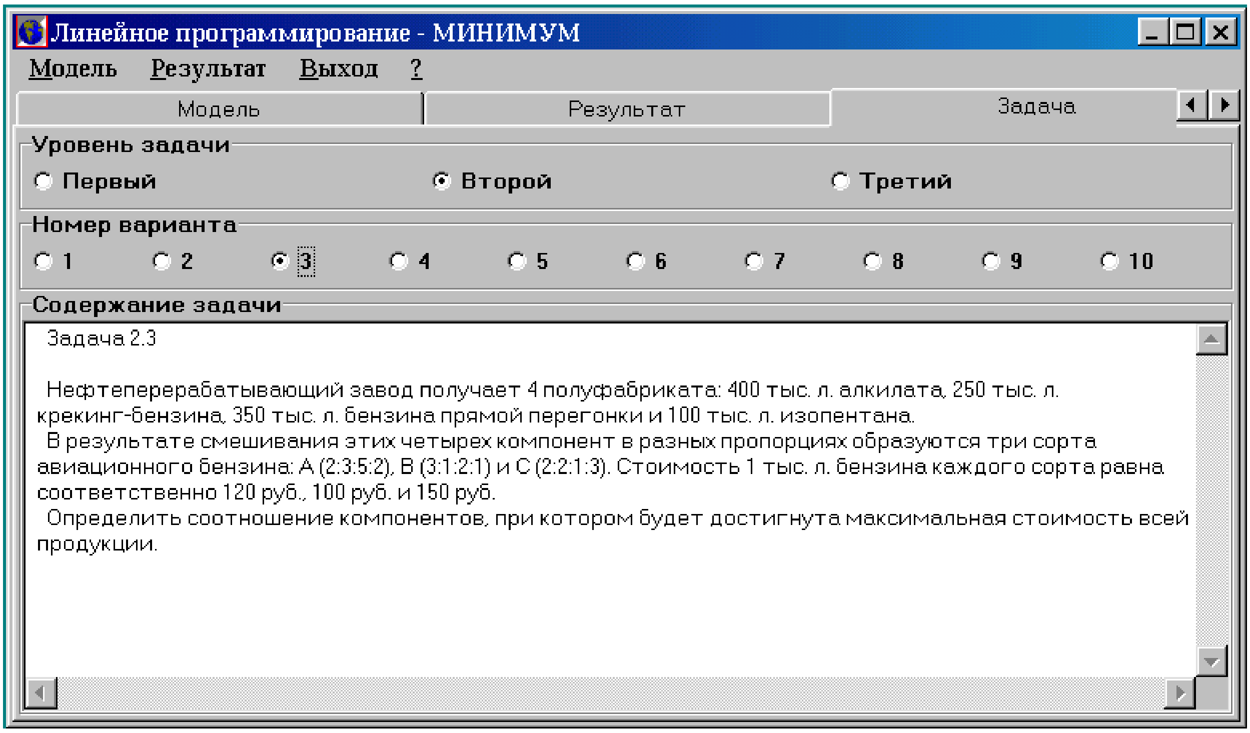

Рис. 4. Страница «Задача»

5.5 Основное меню

Меню состоит из четырех пунктов:

- пункт «Модель» предназначен для выполнения операций с моделями задач;
- пункт «Результат» предназначен для выполнения операций с результатами решения задач;
- с помощью пункта «Выход» можно выйти из программы;
- пункт «?» выводит справку.

Подпункт «Модель->Новая» производит очистку полей ввода модели и осуществляет переход на страницу «Модель».

С помощью подпункта «Модель->Открыть» можно считать модель с диска, которая была сохранена с помощью подпункта «Модель->Сохранить». Файлы моделей имеют формат «*.mdl». Изменение содержимого файла модели приведет к ошибке в программе.

Подпункт «Результат->открыть» выводит на страницу результат текст, сохраненный в текстовом виде. Возможно открытие любого текстового файла и его редактирование.

Действие подпункта «Результат->Вычислить» аналогично соответствующей кнопке на странице.

Подпункт «Результат->Проверить» используется только при выполнении лабораторной работы и предназначен для проверки правильности составления модели задачи по оптимальному значению целевой функции.

5.6 Порядок решения задачи с помощью программного модуля

Для решения задачи необходимо выполнить приведенную последовательность действий.

- Запустить файл Lin_Prog.exe
- Перейти на страницу «Модель»
- Задать количество переменных в целевой функции
- Ввести вектор коэффициентов целевой функции
- Задать количество ограничений задачи
- Ввести матрицу ограничений
- Задать тип ограничений для каждой строки матрицы ограничений
- Ввести столбец свободных членов системы ограничений
- Проверить правильность ввода
- Получить результат решения в автоматическом или пошаговом режиме
- Сохранить результат решения в файл

При вводе модели из файла количество переменных и ограничений выставляются автоматически.

Порядок выполнения работы

1. Ознакомиться с работой программного комплекса на примере контрольной (тестовой) задачи.

2. Подтвердить с помощью комплекса решение предварительно подобранной задачи линейного программирования.

3. Получить задание у преподавателя на решение задач трех уровней.

4. Построить модели двойственных задач и получить решения.

5. Оформить отчет с приведением результатов решения задач с обоснованием построения моделей.

Валерий Израилевич Рейзлин

Методические указания к выполнению лабораторных работ по дисциплине «Методы оптимизации»

Научный редактор

доктор технических наук, профессор В.К.Погребной

Редактор

Подписано к печати
Формат 60x84/16. Бумага офсетная.
Плоская печать. Усл. печ. л. . Уч.- изд. л. .
Тираж 200 экз. Заказ № . Цена свободная.
ИПФ ТПУ. Лицензия ЛТ № 1 от 18.07.94.
Типография ТПУ. 634034, Томск, пр. Ленина, 30.