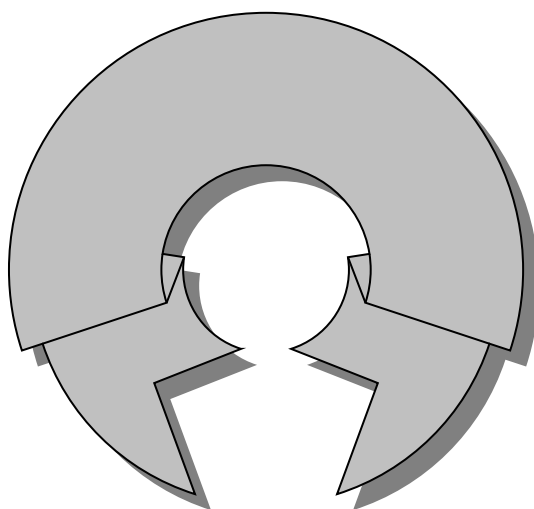


С.Ю. Белецкая

**ТЕХНОЛОГИЯ
АВТОМАТИЗИРОВАННОГО
РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ**



Воронеж 2019

ВВЕДЕНИЕ

При изучении методов оптимального поиска важное значение имеет приобретение практических навыков решения и анализа задач оптимизации с использованием различных программных средств. Однако специализированные системы оптимизации являются разрозненными, недостаточно доступными широкому кругу пользователей, а литература по их применению практически отсутствует. Использование средств EXCEL и MATHCAD обладает рядом преимуществ, связанных с универсальностью данного программного обеспечения, его распространенностью, удобством, доступностью, а также широкими возможностями по решению задач оптимизации различных классов.

В пособии изложен материал, позволяющий представление о возможностях практического использования средств EXCEL и MATHCAD при решении основных классов оптимизационных задач. Пособие состоит из семи глав. Первая глава является вводной и посвящена формализации задач поиска оптимальных вариантов. Во второй главе рассматриваются методы одномерной оптимизации. В третьей, четвертой, пятой и шестой главах обсуждаются вопросы, связанные с решением задач линейного, целочисленного и нелинейного программирования с использованием средств EXCEL. В начале каждой главы дается постановка задачи определенного класса, кратко рассматриваются методы ее решения, излагаются основные приемы построения и типизации соответствующих математических моделей, приводятся практические примеры. Затем последовательно рассматриваются основные этапы решения данных задач на базе EXCEL. Седьмая глава посвящена решению оптимизационных задач с использованием системы MATHCAD. Пособие содержит материал для организации лабораторного практикума.

1. МАТЕМАТИЧЕСКИЕ И АЛГОРИТМИЧЕСКИЕ ОСНОВЫ ПОИСКА ОПТИМАЛЬНЫХ РЕШЕНИЙ

1.1. Обобщенная математическая модель оптимального выбора

Решение широкого круга задач автоматизированного проектирования и управления связано с принятием оптимальных решений, что требует использования соответствующих процедур оптимизации. Под *оптимизацией* будем понимать процедуру определения в соответствии с установленными критериями наилучшего (оптимального) варианта из множества допустимых вариантов. При этом объектами оптимизации могут являться различные технические, производственные, социально-экономические, информационно-телекоммуникационные системы, транспортные и вычислительные сети, механические конструкции, информационные потоки, маршруты и. т. д.

Каждый объект характеризуется набором параметров. При этом различают *внешние*, *внутренние* и *выходные* параметры.

Выходные параметры – величины, характеризующие свойства объекта и определяющие степень выполнения им своего функционального назначения (мощность, коэффициент усиления, производительность, надежность, стоимость и т. д.).

Внешние параметры – величины, характеризующие свойства внешней по отношению к объекту среды и оказывающие влияние на его функционирование (температура окружающей среды, параметры входных сигналов, напряжение источников питания и т. д.).

Внутренние параметры – величины, характеризующие свойства отдельных элементов объекта (геометрические размеры, электрические параметры и т. д.).

Совокупность внутренних и внешних параметров называется *входными* параметрами объекта. При этом входные параметры иг-

рают роль независимых переменных, а выходные параметры являются зависимыми от них величинами. Входные параметры делятся на *управляемые* (варьируемые) и *неуправляемые*. Варьируемые параметры в процессе оптимизации могут изменяться, что вызывает, в свою очередь, изменение выходных параметров. Варьируемые параметры в зависимости от решаемой задачи могут иметь различную интерпретацию. В дальнейшем будем говорить только о варьируемых параметрах объекта и обозначать их следующим образом:

$$X = (x_1, \dots, x_n).$$

Различают задачи *параметрической* и *структурной* оптимизации. *Структурная оптимизация* связана с определением оптимальной структуры объекта (т. е. входящих в него элементов и их взаимосвязей). При *параметрической оптимизации* определяются значения параметров объекта заданной структуры, при которых достигается наилучшее сочетание его свойств. Например, при структурной оптимизации вычислительной системы осуществляется выбор количества и типов оборудования, определение топологии сети и т. д. Параметрическая оптимизация может быть связана с определением оптимальной интенсивности поступления задач на обслуживание. В этом случае соответствующие интенсивности будут являться варьируемыми параметрами при оптимизации.

Решение задачи оптимизации начинается с формирования математической оптимизационной модели. Обобщенно такая модель может быть записана следующим образом:

$$\begin{aligned} f(X) &\rightarrow \min(\max) \\ g_i(X) &\geq (\leq) b_i, \quad i = \overline{1, m} \\ h_k(X) &= d_k, \quad k = \overline{1, s} \\ x_j^{\min} &\leq x_j \leq x_j^{\max}, \quad j = \overline{1, n} \end{aligned} \tag{1.1}$$

Здесь $X = (x_1, \dots, x_n)$ – вектор варьируемых параметров объекта; $f(X)$ – *целевая функция*, т. е. функция, характеризующая наибо-

лее важные и существенные свойства объекта и являющаяся критерием оценки качества каждого из вариантов. Целевая функция в оптимизационной модели может стремиться к максимуму или к минимуму. Ее еще называют *критерием оптимальности, критерием эффективности, показателем качества объекта* и т. д. На основании вычисления значений целевой функции $f(X)$ анализируются и сравниваются между собой различные варианты объектов.

В процессе оптимизации объект должен удовлетворять различным требованиям (конструктивным, техническим, экономическим), которые формализуются в виде системы *ограничений*. Ограничения на варьируемые параметры $x_j^{\min} \leq x_j \leq x_j^{\max}$ называют *прямыми ограничениями*. Здесь $x_j^{\min}$ и $x_j^{\max}$ – нижние и верхние допустимые значения для управляемого параметра x_j , которые зависят от решаемой оптимизационной задачи.

Ограничения вида $g_i(X) \geq (\leq) b_i$, $h_k(X) = d_k$ называют *функциональными ограничениями*. Функциональные ограничения являются какими-либо функциями от варьируемых параметров и формулируются в виде системы равенств и неравенств.

Совокупность функциональных и прямых ограничений образует *допустимую область* или *область допустимых решений* D .

Таким образом, решение задачи оптимизации сводится к определению значений управляемых параметров $X = (x_1, \dots, x_n)$, обеспечивающих оптимальное значение (максимум или минимум) целевой функции $f(X)$ и удовлетворяющих системе ограничений. Такой вектор $X = (x_1, \dots, x_n)$ называется *оптимальным решением*.

Рассмотренная математическая оптимизационная модель является обобщенной и конкретизируется при решении практических задач оптимизации. При этом в зависимости от вида оптимизационной модели задачи оптимизации и методы их решения делятся на классы.

1.2. Классификация задач оптимизации

1. В зависимости от числа управляемых параметров различают задачи *одномерной* и *многомерной* оптимизации. В задачах одномерной оптимизации имеется один варьируемый параметр, а в задачах многомерной оптимизации – несколько параметров.

2. По характеру искомого оптимума различают задачи *локальной* и *глобальной* оптимизации (унимодальные и многоэкстремальные задачи).

3. В зависимости от наличия ограничений выделяют задачи *безусловной* и *условной* оптимизации. В задачах безусловной оптимизации ограничения отсутствуют, и варьируемые параметры могут изменяться в любых пределах.

4. В зависимости от количества критериев оптимальности различают задачи *однокритериальной* (скалярной) и *многокритериальной* (векторной) оптимизации. В задачах многокритериальной оптимизации имеется несколько критериев оптимальности.

5. По виду целевой функции и ограничений различают задачи *линейной* и *нелинейной* оптимизации. В задачах линейной оптимизации целевая функция и все функции-ограничения линейны. Если хотя бы одна из функций является нелинейной, задача относится к классу задач нелинейной оптимизации.

6. По характеру изменения варьируемых параметров различают задачи *непрерывной* и *дискретной* оптимизации. В задачах непрерывной оптимизации параметры изменяются непрерывно в пределах, установленных функциональными ограничениями. В задачах дискретной оптимизации варьируемые параметры принимают дискретные значения.

Частными случаями задач дискретной оптимизации являются задачи *целочисленной* и *булевой* оптимизации. В задачах целочисленной оптимизации варьируемые параметры могут принимать только целочисленные значения. В задачах булевой оптимизации

$x_j \in \{0,1\}$. Если D – дискретное конечное множество, то такая дискретная задача называется *комбинаторной*. Если при оптимизации часть параметров дискретна, а часть имеет непрерывный характер, то рассматривается задача *частично дискретной* (непрерывно-дискретной) оптимизации.

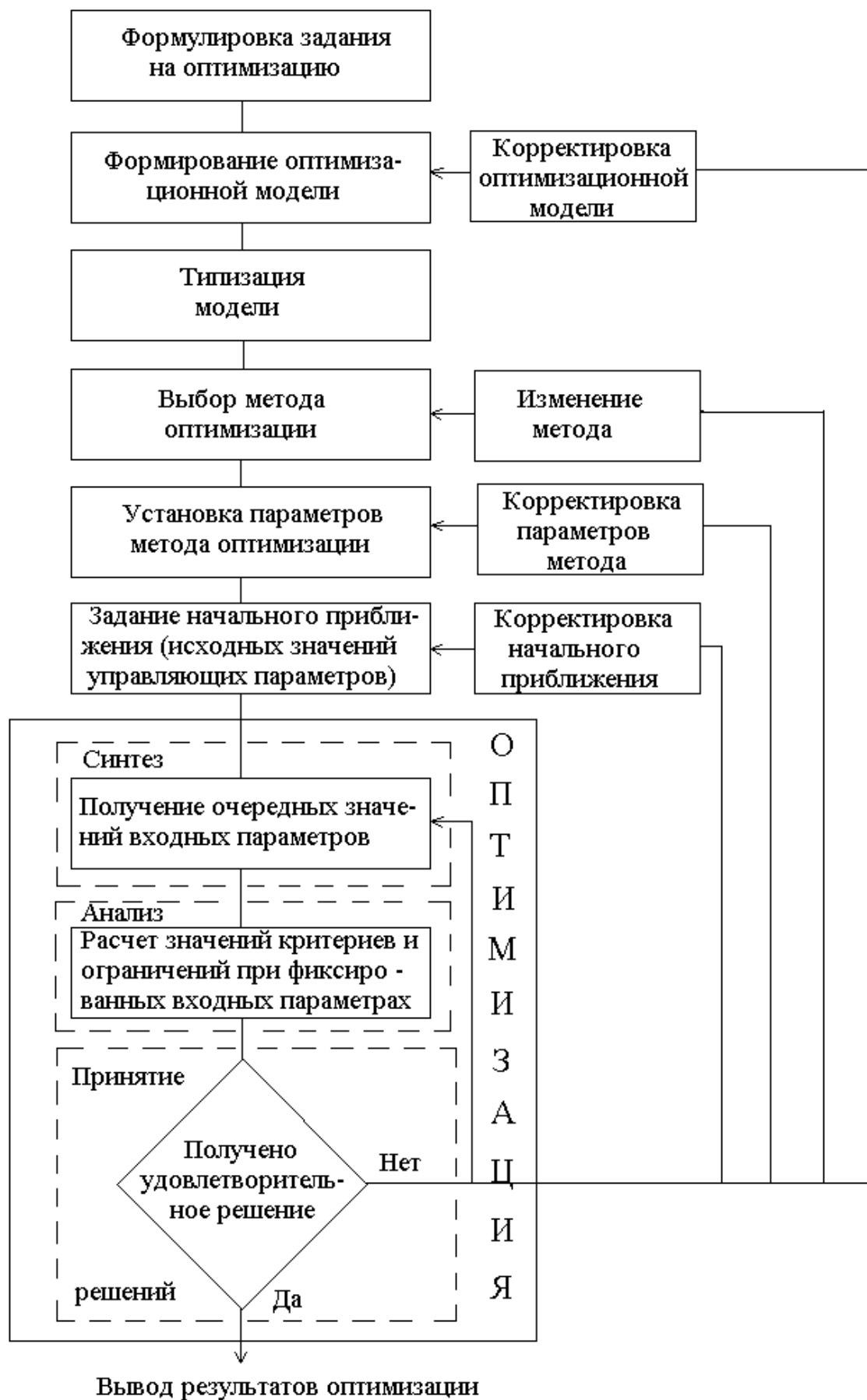
Необходимо заметить, что одна и та же оптимизационная задача может относиться к нескольким типам (например, являться однокритериальной, линейной и непрерывной). Для различных типов оптимизационных задач разработаны соответствующие методы их решения, классификация которых является аналогичной.

1.3. Алгоритмическая схема автоматизированного поиска оптимальных решений

Несмотря на то, что задачи оптимизации решаются в разных предметных областях и характеризуются многообразием постановок и методов решения, можно выделить некоторую обобщенную последовательность типовых этапов поиска оптимальных вариантов (рисунок). Рассмотрим эти этапы более подробно.

На этапе формирования задания на оптимизацию осуществляется содержательная постановка задачи, т. е. определяется набор варьируемых параметров и требования к ним, выбираются критерии оптимальности. Исходной информацией при этом являются данные о назначении, условиях применения и режимах функционирования объекта оптимизации, а также его описание. Эти данные позволяют определить цели оптимизации и в дальнейшем формализовать технические требования к объекту в виде оптимизационной модели.

Формирование оптимизационной модели связано с формализацией содержательной постановки задачи, т. е. с ее математической записью в форме (1.1). При этом необходимо определить взаимосвязи между входными и выходными параметрами и сформулировать критерии оптимальности и функциональные ограничения.



Алгоритмическая схема поиска оптимальных вариантов

Типизация модели связана с ее преобразованием и приведением к типовому виду, допускающему использование стандартных оптимизационных процедур.

Одним из важнейших этапов поиска оптимальных вариантов является рациональный выбор метода оптимизации. Как уже было отмечено в п. 1.2, задачи оптимизации делятся на классы, для каждого из которых разработаны свои методы и алгоритмы решения. Отнесение задачи к определенному классу и определение соответствующего класса методов осуществляется на основании оценки свойств задачи (числа критериев и ограничений, характера изменения параметров и т. д.) и, как правило, сложности не представляет. Однако в рамках каждого класса методов существуют множество различных оптимизационных процедур, выбор которых является достаточно сложной проблемой, так как один и тот же алгоритм при решении различных задач может показывать разную эффективность. Поэтому на практике часто используется многометодный режим решения задач оптимизации.

Далее в зависимости от выбранного метода оптимизации определяются его параметры, а также выбирается начальное приближение. От рационального выбора этих величин также во многом зависят результат и трудоемкость решения оптимизационной задачи.

Сам процесс оптимизации представляет собой итерационную последовательность процедур синтеза, анализа и принятия решений. На этапе синтеза в соответствии с выбранным алгоритмом определяются очередные значения варьируемых параметров (т. е. синтезируется очередной вариант сочетания значений параметров). Этап анализа связан с оценкой полученного варианта, т. е. с вычислением значений критериев и ограничений, соответствующих полученным значениям варьируемых параметров. На этапе принятия решений определяется, является ли данный вариант удовлетворительным. Если получен приемлемый вариант, процесс оптимизации заканчивается, в противном случае – продолжается, т. е. осуществляется пе-

реход к следующей итерации. Обычно в вычислительных процедурах оптимизации проверка качества получаемого варианта осуществляется автоматически. При этом на начальном этапе оптимизации определяется требуемая точность вычислений, при достижении которой алгоритм заканчивает работу.

Если при завершении оптимизационного процесса полученный вариант не устраивает проектировщика, можно выполнить следующие действия:

- изменить начальное приближение;
- изменить параметры метода;
- выбрать другой метод оптимизации;
- скорректировать оптимизационную модель.

После внесения соответствующих изменений оптимизационный процесс продолжается до получения оптимального результата.

Необходимо заметить, что все этапы рассмотренного итерационного процесса, кроме непосредственно оптимизации, выполняются пользователем. Таким образом, решение каждой практической задачи, связанной с поиском оптимальных вариантов, требует индивидуального подхода и во многом зависит от опыта и интуиции проектировщика. Поэтому при разработке современных систем оптимизации большое внимание уделяется вопросам принятия оптимальных решений в интерактивном режиме, когда пользователь имеет возможность оперативно взаимодействовать с ЭВМ на любом этапе решения своей задачи.

2. РЕШЕНИЕ ЗАДАЧ ОДНОМЕРНОЙ ОПТИМИЗАЦИИ

Рассмотрим задачу безусловной оптимизации с одним варьируемым параметром x :

$$f(x) \rightarrow \min_{x \in \mathbb{R}}.$$

Функция $f(x)$ имеет локальный минимум в точке x_0 , если существует некоторая положительная величина δ , такая, что если $|x - x_0| < \delta$, то $f(x) \geq f(x_0)$, т. е. если существует окрестность точки x_0 , такая, что для всех значений x в этой окрестности $f(x)$ больше $f(x_0)$. Функция $f(x)$ имеет глобальный минимум в точке x^* , если для всех x справедливо неравенство: $f(x) \geq f(x^*)$.

На рис. 2.1 представлена функция $f(x)$, которая имеет локальный минимум в точке x_0 и глобальный минимум в точке x^* .

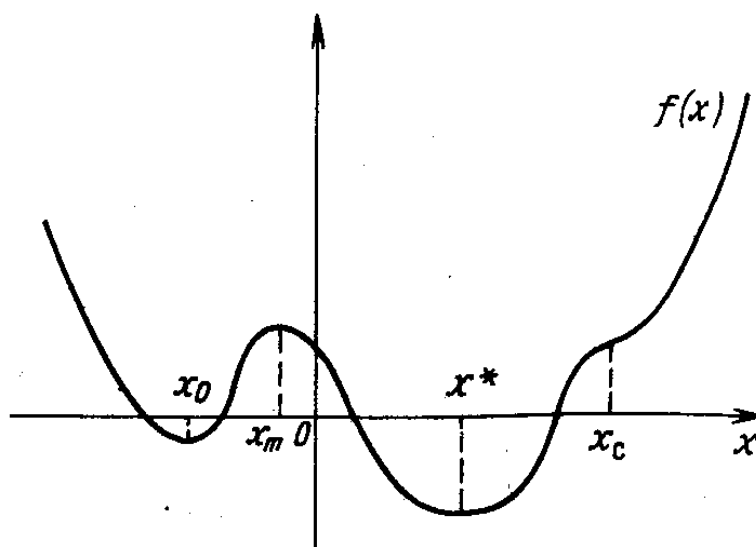


Рис. 2.1

Классический подход к задаче нахождения значений x_0 и x^* состоит в поиске уравнений, которым они должны удовлетворять, т. е. в использовании необходимых и достаточных условий экстремума.

Необходимым условием экстремума функции одной переменной является равенство нулю первой производной данной функции в

точке оптимума. Следовательно, точки x_0 и x^* будут решениями уравнения $f'(x) = 0$.

Достаточное условие экстремума может быть сформулировано следующим образом: если функция $f(x)$ и ее производные непрерывны, то точка x^* является точкой экстремума (максимума или минимума) тогда и только тогда, когда n четное, где n — порядок первой не обращающейся в нуль в точке x^* производной. Если $f^{(n)}(x^*) < 0$, то в точке x^* достигается максимум. Если $f^{(n)}(x^*) > 0$, то в точке x^* достигается минимум.

Непосредственный поиск точек оптимума функции $f(x)$ с использованием необходимых и достаточных условий экстремума является затруднительным, поэтому на практике используются вычислительные методы решения задач одномерной оптимизации. Однако необходимо заметить, что эти методы предназначены для решения задач *унимодального* поиска.

2.1. Метод Ньютона

Метод Ньютона основан на численном решении уравнения:

$$\varphi(x) = f'(x) = 0. \quad (2.1)$$

Разложение функции $\varphi(x)$ в окрестности некоторой точки x_k в ряд Тейлора позволяет переписать уравнение (2.1) в виде:

$$\varphi(x) = \varphi(x_k) + (x - x_k)\varphi'(x_k) + o(|x - x_k|) = 0. \quad (2.2)$$

Отбрасывая в (2.2) малые высшего порядка, получим линейное уравнение относительно неизвестной величины x , решением которого будет

$$x = x_k - \frac{\varphi(x_k)}{\varphi'(x_k)}.$$

Значение аргумента x , определяемое по этой формуле, дает новое приближение корня уравнения (2.1). Таким образом, метод Ньютона отыскания решения уравнения (2.1) описывается следующей итерационной схемой:

$$x_{k+1} = x_k - \frac{\varphi(x_k)}{\varphi'(x_k)} \quad \text{или} \quad x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}. \quad (2.3)$$

Метод Ньютона называют также *методом касательных* из-за его геометрической интерпретации (рис. 2.2).

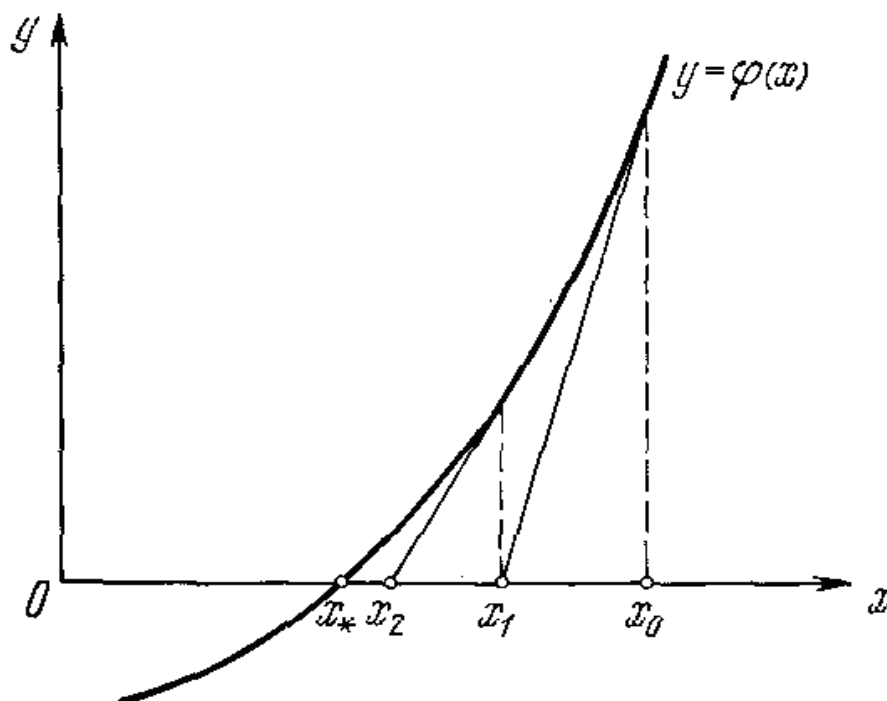


Рис. 2.2.

Проведем касательную к кривой $\varphi(x)$ в точке x_k . Точка пересечения этой касательной с осью абсцисс x_{k+1} , определяется по формуле (2.3). Таким образом, метод Ньютона для решения уравнения $\varphi(x)=0$ состоит в последовательном построении точек пересечения касательных к кривой $\varphi(x)$ с осью абсцисс.

Итерации могут быть продолжены до выполнения условия: $|x_{k+1} - x_k| < \varepsilon$, где ε - требуемая точность.

Основные шаги алгоритма :

1. Задать начальное значение x_0 и точность ε . Положить $k=0$.
2. Найти значения первой и второй производных целевой функции в точке x_k : $f'(x_k)$ и $f''(x_k)$.
3. Определить: $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$.
4. Если $|x_{k+1} - x_k| < \varepsilon$, работа алгоритма заканчивается. В противном случае положить $k=k+1$ и перейти к шагу 2.

2.2. Метод секущих

Метод секущих является комбинацией метода Ньютона и схемы исключения интервалов. 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. 2.3). Таким образом, следующее приближение к стационарной точке x^* определяется по формуле

$$z = R - \frac{f'(R)(R-L)}{f'(R) - f'(L)} \quad (2.4)$$

Если $|R-L| \leq \varepsilon$, 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. Например, в ситуации, изображенной на рис. 2.3, в качестве двух следующих точек должны быть выбраны точки z и R .

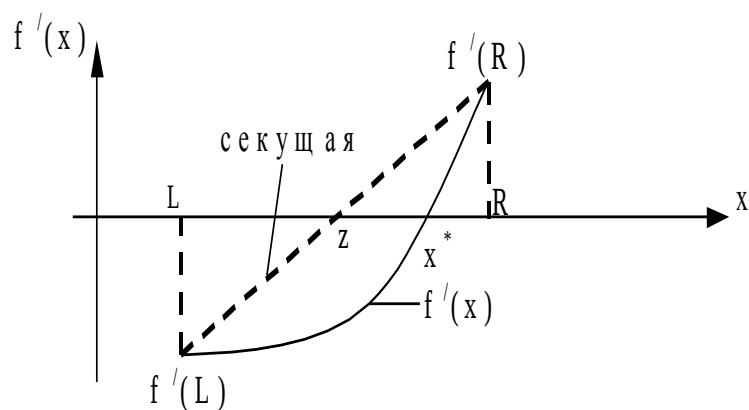


Рис. 2.3.

Основные шаги вычислительной схемы, реализующей метод секущих:

Шаг 1. Задать начальный интервал $(L; R)$, такой, что $f'(L)$ и $f'(R)$ имеют разные знаки, и точность ε .

Шаг 2. Вычислить значения производной $f'(L)$ и $f'(R)$.

Шаг 3. Вычислить $z = R - \frac{f'(R)(R-L)}{f'(R)-f'(L)}$, $f'(L)$. Если $f'(L)$ имеет такой же знак, как $f'(z)$, то $L = z$; если такой же как $f'(R)$, то $R = z$.

Шаг 4. Если $|R-L| > \varepsilon$, возврат к шагу 2, иначе - конечный интервал неопределенности равен $(L; R)$.

2.3. Метод дихотомии

Пусть известен интервал $[a, b]$, содержащий точку минимума, который называется *интервалом неопределенности*. В методе дихотомии этот интервал делится пополам и определяются две точки испытаний x_1 и x_2 по формулам:

$$x_1 = c - \frac{\delta}{2} \quad \text{и} \quad x_2 = c + \frac{\delta}{2}.$$

Здесь c – середина отрезка $[a; b]$: $c = \frac{a+b}{2}$; δ – минимально допустимый сдвиг ($\delta > 0$) между точками испытаний x_1 и x_2 , при ко-

тором можно точно определить знак разности $(f(x_1) - f(x_2))$. При этом параметр δ имеет такое значение, что если $|x_1 - x_2| \geq \delta$, то $f(x_1) = f(x_2)$ только в том случае, когда x_1 и x_2 находятся по разные стороны от точки локального минимума x^* .

Следовательно, точки x_1 и x_2 располагаются симметрично относительно середины априорного интервала неопределенности $[a; b]$. Из свойства унимодальности минимизируемой функции $f(x)$ после отбрасывания подынтервала, где точка локального минимума x^* отсутствует, для дальнейшего рассмотрения остается один из текущих интервалов неопределенности $\left[a; c + \frac{\delta}{2} \right]$ или $\left[c - \frac{\delta}{2}; b \right]$, длина которого определяется выражением:
$$L = \frac{b - a}{2} + \frac{\delta}{2}.$$

При этом возможна одна из следующих ситуаций:

а) если $f(x_1) < f(x_2)$ (рис. 2.4, а), то отрезком локализации точки x^* является подынтервал $[a; x_2]$, а отрезок $[x_2; b]$ может быть исключен из дальнейшего рассмотрения;

б) если $f(x_1) > f(x_2)$ (рис. 2.4, б), то отрезком локализации точки x^* является подынтервал $[x_1; b]$, а из рассмотрения исключается отрезок $[a; x_1]$;

в) если $f(x_1) = f(x_2)$ (рис. 2.4, в), то отрезком локализации точки x^* является подынтервал $[x_1; x_2]$, а отрезки $[a; x_1]$ и $[x_2; b]$ могут быть исключены из дальнейшего рассмотрения.

Следующая пара испытаний, разнесенная между собой на величину минимально допустимого сдвига δ , проводится симметрично относительно середины текущего интервала неопределенности, полученного на предыдущем шаге.

Процесс поиска заканчивается, если $|x_{k+1} - x_k| < \varepsilon$, где ε - требуемая точность вычисления оптимума, которая является входным параметром алгоритма.

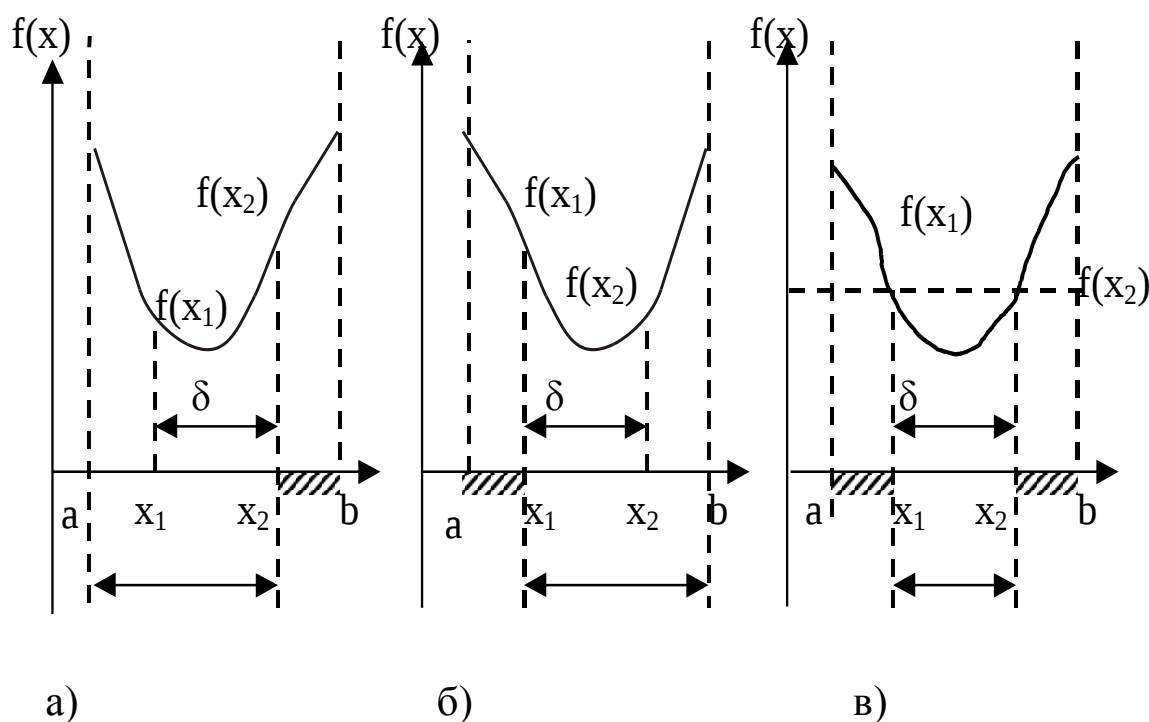


Рис. 2.4.

Основные шаги алгоритма :

1. Задать интервал $[a; b]$, минимально допустимый сдвиг $\delta > 0$, точность ε .

2. Определить $x_1 = \frac{a + b - \delta}{2}$ и $x_2 = \frac{a + b + \delta}{2}$.

3. Вычислить значения функции $f_1 = f(x_1)$ и $f_2 = f(x_2)$.

4. Если $f_1 < f_2$, то $b = x_2$;

если $f_1 > f_2$, то $a = x_1$;

если $f_1 = f_2$, то $a = x_1$ и $b = x_2$.

5. Если $|b - a| < \varepsilon$, работа алгоритма заканчивается. В противном случае необходимо вернуться к шагу 2.

2.4. Метод Фибоначчи

Предположим, что имеется интервал неопределенности $[x_1; x_3]$ и известна точка x_2 внутри этого интервала. В методе

Фибоначчи следующая точка x_4 помещается внутри текущего интервала симметрично относительно уже находящейся там точки. Затем одна из частей интервала неопределенности отбрасывается.

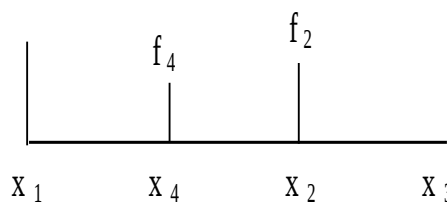
При этом если $f(x_2) = f_2$ и $f(x_4) = f_4$, то можно рассмотреть четыре случая (рис. 2.5):

а) $x_4 < x_2$

$f_4 < f_2$

новый интервал (x_1, x_2)

содержащий точку x_4

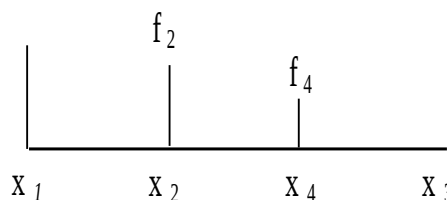


б) $x_4 > x_2$

$f_4 < f_2$

новый интервал (x_2, x_3)

содержащий точку x_4

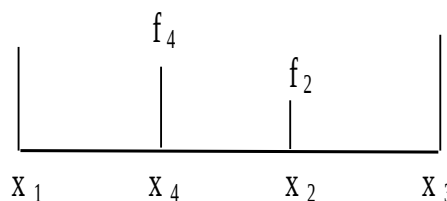


в) $x_4 < x_2$

$f_4 > f_2$

новый интервал (x_4, x_3)

содержащий точку x_2



г) $x_4 > x_2$

$f_4 > f_2$

новый интервал (x_1, x_4)

содержащий точку x_2

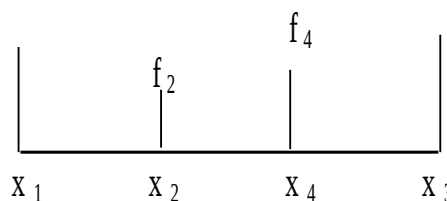


Рис. 2.5.

Таким образом, для работы метода Фибоначчи необходимо найти положение первой точки, после чего итерационная процедура может быть продолжена с использованием рассмотренного правила симметрии.

Пусть L_1 – длина начального интервала. При этом первая точка помещается на расстоянии L_2 от одного из концов начального интервала. Величина L_2 определяется следующим образом:

$$L_2 = \frac{F_{n-1}}{F_n} L_1 + \frac{(-1)^n \varepsilon}{F_n}. \quad (2.5)$$

Здесь n – число испытаний, которое задается перед началом работы алгоритма; F_n и F_{n-1} – числа Фибоначчи. Последовательность чисел Фибоначчи определяется следующим образом:

$$F_0 = 1, F_1 = 1, F_k = F_{k-1} + F_{k-2} \quad k=2, \dots, n$$

Таким образом, каждое последующее число Фибоначчи, начиная с третьего, определяется как сумма двух предыдущих: 1, 1, 2, 3, 5, 8, ...

После того как найдено положение первой точки, числа Фибоначчи больше не нужны. Значение ε задается пользователем.

Основные шаги алгоритма :

1. Задать число испытаний n , параметр ε и начальный интервал неопределенности $[a;b]$. Положить $x_1=a$ и $x_3=b$.

2. Найти положение первой точки по правилу:

$$x_2 = a + \frac{F_{n-1}}{F_n} (b - a) + \frac{(-1)^n \varepsilon}{F_n}.$$

3. Точку x_4 поместить по правилу симметрии на расстоянии x_2 от другого конца отрезка: $x_4 = x_1 - x_2 + x_3$.

4. Вычислить значения функции $f(x_2) = f_2$ и $f(x_4) = f_4$.

5. Установить новые границы интервала неопределенности:

если $x_4 < x_2$ и $f_4 < f_2$, то $x_3 = x_2$, $x_2 = x_4$, $f_2 = f_4$;

если $x_4 > x_2$ и $f_4 < f_2$, то $x_1 = x_2$, $x_2 = x_4$, $f_2 = f_4$;

если $x_4 < x_2$ и $f_4 > f_2$, то $x_1 = x_4$;

если $x_4 > x_2$ и $f_4 > f_2$, то $x_3 = x_4$.

6. Вернуться к шагу 3. Выполнять до тех пор, пока число испытаний не станет равным n .

2.5. Метод золотого сечения

Не всегда можно заранее определить, сколько раз придется вычислять функцию. В методе Фибоначчи это нужно знать для определения L_2 , т.е. положения начальной точки.

Метод «золотого сечения» почти столь же эффективен, как и метод Фибоначчи, однако при этом не требуется знать n – количество вычислений функции, определяемое вначале. После того как выполнено j вычислений, исходя из тех же соображений, что и ранее, записываем $L_{j+1} = L_j + L_{j+1}$. Однако если n неизвестно, то мы не можем использовать условие $L_{n+1} = 2L_n - \varepsilon$. Если отношение последующих интервалов будет постоянным, т.е.

$$\frac{L_{j+1}}{L_j} = \frac{L_j}{L_{j+1}} = \frac{L_{j+2}}{L_{j+1}} = \dots = \tau, \quad (2.6)$$

то $\frac{L_{j+1}}{L_j} = 1 + \frac{L_{j+1}}{L_j}$, т.е. $\tau = 1 + 1/\tau$.

Таким образом, $\tau^2 - \tau - 1 = 0$, откуда $\tau = (1 + \sqrt{5})/2 \approx 1.618033989$.

Тогда $\frac{L_{j+1}}{L_j} = \tau^2$, $\frac{L_{j+2}}{L_{j+1}} = \tau^3$ и т.д. Следовательно, $L_n = \frac{L_1}{\tau^{n-1}}$.

В результате анализа двух рассмотренных значений функций будет определен тот интервал, который должен исследоваться в дальнейшем. Этот интервал будет содержать одну из предыдущих точек и следующую точку, помещаемую симметрично ей. Первая точка находится на расстоянии L_1/τ от одного конца интервала, вторая – на таком же расстоянии от другого. Поскольку $\lim_{n \rightarrow \infty} \frac{F_{n-1}}{F_n} = \frac{1}{n}$, то поиск методом «золотого сечения» является предельной формой поиска методом Фибоначчи. Название «золотое сечение» произошло

от названия отношения в уравнении (2.6). Видно, что L_{j-1} делится на две части так, что отношение целого к большей части равно отношению большей части к меньшей, т.е. равно так называемому «золотому отношению».

Таким образом, если ищется интервал $(x_0; x_3)$ и имеются два значения функции f_1 и f_2 в точках x_1 и x_2 , то следует рассмотреть два случая (рис. 2.6).

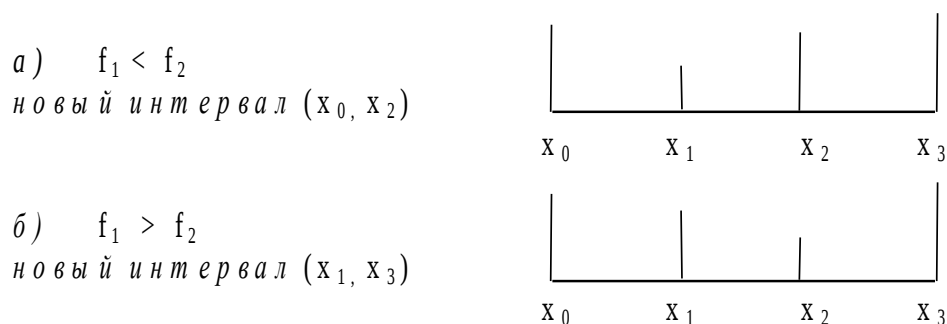


Рис. 2.6.

Основные шаги алгоритма:

1. Задать интервал $(A; B)$, точность ε .
2. Положить $x_0 = A$ и $x_3 = B$. Определить первую точку $x_1 = B - \frac{L_1}{\tau}$,
где $L_1 = (x_3 - x_0)$; $x_2 = x_0 + x_3 - x_1$.
3. Вычислить значения функции в точках x_1 и x_2 — f_1 и f_2 .
4. Установить новые значения координат точек в зависимости от соотношения f_1 и f_2 :
 - а) если $f_1 < f_2$, то $I = x_2 - x_0$; $x_3 = x_2$; $x_2 = x_1$; $x_1 = x_0 + \frac{I}{\tau}$; $f_2 = f_1$.
 - б) если $f_1 > f_2$, то $I = x_3 - x_1$; $x_0 = x_1$; $x_1 = x_2$; $x_2 = x_0 + \frac{I}{\tau}$; $f_1 = f_2$.
5. Если $I > \varepsilon$, возврат к шагу 3. Иначе, конечным результатом поиска считается точка x_1 .

2.6. Лабораторная работа № 1

ИЗУЧЕНИЕ МЕТОДОВ ОДНОМЕРНОЙ ОПТИМИЗАЦИИ

Цель работы: Изучение методов одномерной оптимизации, формирование практических навыков программной реализации алгоритмов одномерной оптимизации.

Лабораторные задания.

1. По п.2.1 - 2.5 настоящего пособия ознакомиться с методами одномерного поиска и алгоритмами их работы.
2. В соответствии с указанием преподавателя выбрать вариант задания и оптимизируемую функцию.
3. Программно реализовать указанный в задании метод поиска применительно к данной функции.

Задание на лабораторную работу выбирается из вариантов, представленных ниже. Требуется написать программу, осуществляющую поиск минимума функции по одному из рассмотренных алгоритмов: Ньютона (вариант 1,2), дихотомии (вариант 3,4,5), Фибоначчи (вариант 6,7,8), золотого сечения (вариант 9.10). Минимизируемая функция выбирается из таблицы. Номер строки таблицы (номер функции) соответствует варианту. В зависимости от метода из той же строки выбираются необходимые параметры – для метода Фибоначчи количество вычислений и константа различимости, для остальных методов точность.

Отчет по лабораторной работе должен содержать:

1. Название лабораторной работы и ее цель.
2. Краткие теоретические сведения по всем методам поиска.
3. Блок-схему выбранного алгоритма поиска.
4. Листинг реализованной программы.
5. Контрольный пример работы программы.
6. Выводы.

№	Функция	Интервал	Параметры
	1	2	4
1	$2x^2 + 3e^{-x}$	[0;1]	Точность: 0,000001
2	$x^4 - 14x^3 + 60x^2 - 70x$	[5;7]	Точность: 0,000001
3	$e^{-x} - \cos(x)$	[0;1]	Точность: 0,000001
4	$2x^2 - e^x$	[-1;1]	Точность: 0,000001
5	$-(e^{-x} \cdot \ln(x))$	[1;3]	Точность: 0,000001
6	$-(e^{-x} x^3)$	[2;4]	Количество вычислений 10 Константа различимости $\varepsilon = 0,0000001$
7	$x^2 \cdot \sin(0,5x)$	[9;11]	Количество вычислений 14 Константа различимости $\varepsilon = 0,0000001$
8	$\ln(x) + 3 \cos(x)$	[2;4]	Количество вычислений 12 Константа различимости $\varepsilon = 0,0000001$
9	$-\sqrt{x} \sin\left(\frac{x^2}{100}\right)$	[10;15]	Точность: 0,000001
10	$\cos(x^{0,6})$	[4;8]	Точность: 0,000001

3. ЛИНЕЙНАЯ ОПТИМИЗАЦИЯ

3.1. Модели линейной оптимизации

Задачи линейной оптимизации заключаются в определении оптимального (максимального или минимального) значения линейной целевой функции при линейных ограничениях.

$$\begin{aligned} F &= \sum_{j=1}^n c_j x_j \rightarrow \max(\min) , \\ \sum_{j=1}^n a_{ij} x_j &\leq (\geq, =) b_i, i = \overline{1, m}, \\ x_k &\geq 0, k = \overline{1, s}, s \leq n . \end{aligned} \quad (3.1)$$

Здесь $x_1, \dots, x_n$ – варьируемые параметры модели.

При разработке вычислительных методов решения задач линейной оптимизации особое значение имеет *каноническая форма* задачи:

$$\begin{aligned} F &= \sum_{j=1}^n c_j x_j \rightarrow \min , \\ \sum_{j=1}^n a_{ij} x_j &= b_i, i = \overline{1, m}, \\ x_i &\geq 0, i = \overline{1, n}. \end{aligned} \quad (3.2)$$

Каноническая форма характеризуется следующими условиями:

1. Целевая функция подлежит минимизации.
2. Все переменные должны быть неотрицательны.
3. Правые части всех ограничений задачи должны быть неотрицательны.
4. Все ограничения задачи должны быть записаны в виде равенств.

Рассмотрим основные приемы построения математических моделей для некоторых типовых задач линейного программирования.

1. Задача определения производственного плана.

Для изготовления n видов продукции $A_1, \dots, A_n$ необходимы ресурсы $S_1, \dots, S_m$ (сырье, рабочая сила, оборудование и т. д.). Заданы значения a_{ij} , которые характеризуют количество ресурса S_i , необходимого для выпуска единицы продукции A_j . Запас каждого ресурса S_i ограничен и равен b_i . От реализации единицы продукции A_j может быть получена прибыль c_j . Необходимо так составить производственный план (т. е. определить, сколько продукции каждого вида необходимо выпустить), чтобы общая прибыль от производства всей продукции была максимальной.

Исходные данные задачи удобно представлять в виде таблицы (табл. 3.1):

Таблица 3.1

Ресурсы	Виды продукции				Запасы ресурсов
	A_1	A_2	...	A_n	
S_1	a_{11}	a_{12}	...	a_{1n}	b_1
S_2	a_{21}	a_{22}	...	a_{2n}	b_2
...	...	...	...	...	...
S_m	a_{m1}	a_{m2}	...	a_{mn}	b_m
Прибыль	c_1	c_2	...	c_n	

Обозначим через x_j количество продукции A_j , которое необходимо выпустить по плану. Тогда математическая модель данной задачи имеет следующий вид:

$$c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \max;$$

$$\sum_{j=1}^n a_{ij}x_j \leq b_i, \quad i = \overline{1, m};$$

$$x_j \geq 0, \quad j = \overline{1, n}.$$

Здесь целевая функция – это общая прибыль от производства всей продукции. Каждое i -е ограничение характеризует общее количество ресурса S_i , которое не должно превышать величины b_i . Кроме ограничения по ресурсам, в условия задачи (а, следовательно, и в ее математическую модель) могут вводиться дополнительные ограничения на планируемый выпуск продукции (ограничения по ассортименту, условия комплектности и т. д.).

Пример 3.1. Требуется определить, в каком количестве необходимо выпускать продукцию четырех типов П1, П2, П3, П4, для изготовления которой требуются ресурсы трех видов: трудовые ресурсы, сырье, финансы. Нормы расхода ресурсов каждого вида для выпуска единицы продукции, а также прибыль, получаемая от реализации единицы каждого типа продукции, приведены в табл. 3.2. Количество расходуемых ресурсов не должно превышать имеющихся запасов.

Таблица 3.2

Ресурсы	Виды продукции				Запасы ресурсов
	П1	П2	П3	П4	
Трудовые	3	1	2	4	440
Сырье	1	8	6	2	200
Финансы	1	4	7	2	320
Прибыль	7	3	6	12	

Математическая модель для решения данной задачи будет иметь следующий вид:

$$F=7x_1+3x_2+6x_3+12x_4 \rightarrow \max;$$

$$3x_1+x_2+2x_3+4x_4 \leq 440;$$

$$x_1+8x_2+6x_3+2x_4 \leq 200;$$

$$x_1+4x_2+7x_3+2x_4 \leq 320;$$

$$x_j \geq 0, \quad j=\overline{1,4}.$$

2. Задача о смесях.

Имеется набор компонентов $A_1, \dots, A_n$, при сочетании которых в разных пропорциях образуются различные смеси. В состав каждого компонента входят вещества $P_1, \dots, P_m$. Через a_{ij} обозначим количество вещества P_i в единице компонента A_j . В смеси необходимо обеспечить содержание вещества P_i в количестве не менее b_i). Цена единицы компонента A_j равна c_j . Необходимо определить состав смеси (количество компонентов каждого вида), цена которой будет минимальной. Исходные данные задачи представлены в табл. 3.3:

Таблица 3.3

Вещества	Компоненты смеси				Минимально необходимое количество веществ
	A_1	A_2	$\dots$	A_n	
P_1	a_{11}	a_{12}	$\dots$	a_{1n}	b_1
P_2	a_{21}	a_{22}	$\dots$	a_{2n}	b_2
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
P_m	a_{m1}	a_{m2}	$\dots$	a_{mn}	b_m
Цена	c_1	c_2	$\dots$	c_n	

Обозначим через x_j количество компонента A_j , которое необходимо включить в смесь. Тогда математическая модель данной задачи имеет следующий вид:

$$c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \min;$$

$$\sum_{j=1}^n a_{ij}x_j \geq b_i, i = \overline{1, m};$$

$$x_j \geq 0, j = \overline{1, n}.$$

Здесь целевая функция характеризует цену смеси, которая должна быть минимальной. Каждое i -е ограничение характеризует

общее количество вещества P_i в смеси. Кроме ограничений по содержанию отдельных веществ в смеси, в задаче могут быть использованы ограничения по имеющимся запасам отдельных компонентов или по предельным нормам их включения в смесь. Могут задаваться также пропорции, в которых некоторые из компонентов должны входить в состав смеси.

3. Задача о комплектах.

Пусть планируется производство комплектных изделий. Каждое изделие содержит m видов деталей, причем деталей первого вида – k_1 единиц, деталей второго вида – k_2 единиц, деталей j -го вида – k_j единиц. Изделия могут изготавливаться n различными исполнителями. Пусть в единицу времени i -й исполнитель может изготовить a_{ij} элементов j -го вида. Сдаче подлежат только комплектные изделия. Каждый i -й исполнитель работает не более b_i часов. Определить такой план загрузки исполнителей, чтобы общее число выпускаемых комплектных изделий было максимальным.

Обозначим через x_{ij} время изготовления деталей j -го вида i -м исполнителем. Тогда общее число деталей j -го вида, изготавливаемых всеми исполнителями, определится в виде

$$D_j = a_{1j}x_{1j} + \dots + a_{nj}x_{nj}, \quad \forall j = \overline{1, m}.$$

Общее количество изделий, для выпуска которых хватит деталей j -го типа при условии, что k_j единиц расходуется на одно изделие равно

$$I_j = \frac{D_j}{k_j} = \frac{a_{1j}x_{1j} + \dots + a_{nj}x_{nj}}{k_j}.$$

Тогда общее число комплектов определится следующим образом:

$$\min_{j=1, m} I_j$$

В результате математическую модель задачи можно сформулировать в виде:

$$\begin{aligned} \max_{x_{ij}} \left(\min_{j=\overline{1,m}} I_j \right) &= \max_{x_{ij}} \left(\min_{j=\overline{1,m}} \frac{a_{1j}x_{1j} + \dots + a_{nj}x_{nj}}{k_j} \right); \\ x_{i1} + \dots + x_{im} &\leq b_i, \quad i = \overline{1,n}; \\ x_{ij} &\geq 0, \quad i = \overline{1,n}, \quad j = \overline{1,m}. \end{aligned}$$

Для формализации данной постановки в виде задачи линейного программирования введем новую переменную:

$$y = \min_{j=\overline{1,m}} \frac{\sum_{i=1}^n a_{ij}x_{ij}}{k_j},$$

что равносильно записи

$$y \leq \frac{\sum_{i=1}^n a_{ij}x_{ij}}{k_j} \quad \forall j = \overline{1,m}.$$

Тогда окончательный вид математической модели задачи:

$$\begin{aligned} &y \rightarrow \max \\ &\left\{ \begin{array}{l} \sum_{i=1}^n a_{ij}x_{ij} - yk_j \geq 0, \quad j = \overline{1,m} \\ \sum_{j=1}^m x_{ij} \leq b_i, \quad i = \overline{1,n} \\ x_{ij} \geq 0, \quad i = \overline{1,n}, \quad j = \overline{1,m} \end{array} \right. \end{aligned}$$

4. Задача о раскрое.

Для изготовления заготовок $D_1, \dots, D_m$ имеется n способов раскроя материала $A_1, \dots, A_n$. При этом a_{ij} – количество заготовок i -го типа, получаемых из единицы материала при способе A_j . Имеется D единиц материала. При раскрое единицы материала по способу A_j имеются отходы площадью c_j . Необходимо произвести b_i заготовок i -го типа и выполнить план с минимизацией отходов.

Таблица 3.4

Виды заготовок	Способы раскроя				План производства
	A_1	A_2	...	A_n	
D_1	a_{11}	a_{12}	...	a_{1n}	b_1
D_2	a_{21}	a_{22}	...	a_{2n}	b_2
...	...	...	...	...	...
D_m	a_{m1}	a_{m2}	...	a_{mn}	b_m
ОТХОДЫ	c_1	c_2	...	c_n	

Пусть x_j – количество единиц материала, раскраиваемого по j -му способу.

Математическая модель задачи запишется следующим образом:

$$\begin{aligned}
 & \sum_{j=1}^n c_j x_j \rightarrow \min; \\
 & \sum_{j=1}^n a_{ij} x_j = b_i; \quad i = \overline{1, m}; \\
 & \sum_{j=1}^n x_j = D; \\
 & x_j \geq 0, \quad j = \overline{1, n}, \quad x_j - \text{целые}.
 \end{aligned}$$

3.2. Методы решения задач линейной оптимизации

3.2.1. Симплекс-метод

Симплекс-метод является основным методом решения задач линейной оптимизации. Перед его использованием задачу необходимо привести к канонической форме записи. При приведении ЗЛП к канонической форме используются следующие *основные приемы*:

1. Для перехода от задачи максимизации к задаче минимизации целевую функцию необходимо умножить на (-1) .

2. Переменные, на которые не наложено ограничение неотрицательности, представляются в виде разности двух неотрицательных переменных

$$x_j = u_j - w_j, \quad \text{где } u_j, w_j \geq 0.$$

Такая подстановка делается во всех ограничениях, где есть x_j , а также в выражении для целевой функции.

3. Переход к ограничениям с неотрицательными правыми частями осуществляется умножением левой и правой частей ограничений с отрицательными правыми частями на (-1) . При этом знаки соответствующих неравенств меняются на противоположные.

4. Переход от ограничений-неравенств к ограничениям-равенствам осуществляется путем введения дополнительных неотрицательных переменных. При этом, если знак неравенства $\leq$, дополнительная переменная прибавляется к левой части ограничения, а если $\geq$, то вычитается. Таким образом, ограничение-неравенство

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i$$

преобразуется в ограничение-равенство

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n + x_{n+1} = b_i \quad (x_{n+1} \geq 0),$$

а ограничение-неравенство

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \geq b_i,$$

в ограничение-равенство

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n - x_{n+1} = b_i \quad (x_{n+1} \geq 0).$$

В каждое неравенство вводится своя дополнительная переменная. Число вводимых дополнительных неотрицательных переменных равно числу преобразуемых неравенств.

Рассмотрим задачу линейной оптимизации в канонической форме:

$$\begin{aligned} F &= \sum_{j=1}^n c_j x_j \rightarrow \min \\ \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = \overline{1, m} \\ x_j &\geq 0, \quad j = \overline{1, n} \end{aligned}$$

Вектор $X = (x_1, \dots, x_n)$, удовлетворяющий системе ограничений задачи линейной оптимизации, называется *допустимым решением* или *планом*.

План $x^* = (x_1^*, \dots, x_n^*)$, при котором целевая функция принимает оптимальное значение, называется *оптимальным*.

Приведем задачу линейной оптимизации к виду:

[illegible]

Переменные, которые входят только в одно уравнение системы ограничений с коэффициентом 1, называют *базисными переменными* (в рассматриваемом примере это переменные $x_1 \dots x_m$). Остальные переменные называются *свободными*. Тогда приравнивая базисные переменные соответствующим правым частям ограничений,

а свободные переменные нулю, получим опорный план $X = (b_1, \dots, b_m, 0, \dots, 0)$, называемый *опорным планом* задачи линейной оптимизации.

Симплексный метод решения задачи линейной оптимизации основан на переходе от одного опорного плана к другому, при котором значение целевой функции убывает.

Для удобства расчетов в симплексном методе все данные по задаче заносят в симплекс-таблицу:

Таблица 3.5

x_B	c_B	b	c_1	$\dots$	c_n
			x_1	$\dots$	x_n
x_{B1}	c_{B1}	b_1	a_{11}	$\dots$	a_{1n}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
x_{Bm}	c_{Bm}	b_m	a_{m1}	$\dots$	a_{mn}
F			Δ_1	$\dots$	Δ_n

В верхней строке таблицы записываются коэффициенты при всех переменных в целевой функции. В первом столбце таблицы записываются базисные переменные в той последовательности, в которой они входят в систему ограничений, во втором – коэффициенты целевой функции при базисных переменных, в третьем – правые части всех ограничений, в последующих столбцах – коэффициенты при соответствующих переменных в системе ограничений. В нижней строке таблицы записываются оценки по каждой переменной,

определяемые следующим образом: $\Delta_j = \sum_{i=1}^m c_{Bi} a_{ij} - c_j$. Очевидно,

что для базисных переменных оценки равны нулю. Под столбцом правых частей обычно записывается текущее значение целевой функции.

На каждой итерации симплекс-метода осуществляется вывод из базиса какой-либо переменной и включение в него другой пере-

менной с соответствующим пересчетом элементов таблицы. Перед решением задачи ее необходимо привести к канонической форме.

Основные шаги симплекс - метода:

1. Определение начального опорного плана.
2. Составление симплекс-таблицы.

3. Вычисление оценок $\Delta_j = \sum_{i=1}^m c_{Bi} a_{ij} - c_j$.

4. Анализ оценок.

4.1. Если $\Delta_j \leq 0 \quad \forall j = \overline{1, n}$, то получено оптимальное решение.

4.2. Если существует хотя бы одна оценка $\Delta_j > 0$, для которой $a_{ij} < 0 \quad \forall j = \overline{1, m}$, то целевая функция не ограничена снизу на множестве допустимых решений (ЗЛП не имеет решения).

4.3. Из всех оценок $\Delta_j > 0$ выбирается максимальная

$$\Delta_k = \max_{j: \Delta_j > 0} \Delta_j$$

Переменная x_k , которой соответствует максимальная оценка, становится на текущей итерации базисной, а k -й столбец объявляется *ведущим столбцом*.

5. Определение *ведущей строки*. Для этого находятся попарные отношения правых частей ограничений к положительным элементам ведущего столбца и среди них выбирается минимальное:

$$\frac{b_s}{a_{sk}} = \min_{i: a_{ik} > 0} \frac{b_i}{a_{ik}}$$

Строка s объявляется ведущей строкой. Элемент a_{sk} , находящийся в симплекс-таблице на пересечении s -й строки и k -го столбца, становится ведущим элементом.

6. Пересчет элементов симплекс-таблицы. При этом элементы ведущей строки $a_{s1}, \dots, a_{sn}, b_s$ делятся на ведущий элемент a_{sk} . Пересчет остальных элементов удобно осуществлять по правилу прямоугольника.

	k-й столбец		j-й столбец
s-я строка	<u>a_{sk}</u>		a_{sj}
i-я строка	a_{ik}		a_{ij}

Пусть a_{sk} – ведущий элемент, и необходимо пересчитать элемент a_{ij} . Мысленно соединяем эти элементы и достраиваем конструкцию до прямоугольника. Элемент a_{ij} пересчитывается следующим образом:

$$a'_{ij} = \frac{a_{sk}a_{ij} - a_{ik}a_{sj}}{a_{sk}}$$

При этом сначала вычисляется произведение ведущего элемента на пересчитываемый, из него вычитается произведение элементов противоположной диагонали и результат делится на ведущий элемент. По аналогичной схеме пересчитываются и правые части ограничений:

$$b'_i = \frac{a_{sk}b_i - a_{ik}b_s}{a_{sk}}$$

7. Переход к шагу 3.

По окончательной симплекс-таблице оптимальное решение определяется следующим образом: базисные переменные приравниваются соответствующим правым частям ограничений, остальные переменные равны нулю.

Пример 3.2. Решим симплекс-методом задачу определения оптимального производственного плана из примера 3.1. В примере 3.1 была представлена математическая модель данной задачи. Приведем ее к канонической форме:

$$F = -7x_1 - 3x_2 - 6x_3 - 12x_4 \rightarrow \min;$$

$$3x_1 + x_2 + 2x_3 + 4x_4 + x_5 = 440;$$

$$x_1 + 8x_2 + 6x_3 + 2x_4 + x_6 = 200;$$

$$x_1 + 4x_2 + 7x_3 + 2x_4 + x_7 = 320;$$

$$x_j \geq 0, \quad j = \overline{1,7}.$$

Базисными переменными в данном случае будут переменные x_5, x_6, x_7 . Решение задачи иллюстрируется симплексной табл. 3.6. Ведущие элементы на каждой итерации в таблице выделяются.

Таблица 3.6

ХБ	СБ	В	-7	-3	-6	-12	0	0	0
			x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_5	0	440	3	1	2	4	1	0	0
x_6	0	200	1	8	6	<u>2</u>	0	1	0
x_7	0	320	1	4	7	2	0	0	1
0			7	3	6	12	0	0	0
x_5	0	40	<u>1</u>	-15	-10	0	1	-2	0
x_4	-12	100	1/2	4	3	1	0	1/2	0
x_7	0	120	0	-4	1	0	0	-1	1
-1200			1	-45	-30	0	0	-6	0
x_1	-7	40	1	-15	-10	0	1	-2	0
x_4	-12	80	0	23/2	8	1	-1/2	3/2	0
x_7	0	120	0	-4	1	0	0	-1	1
-1240			0	-30	-20	0	-1	-4	0

Так как после третьей итерации все оценки $\Delta_j \leq 0$, получено оптимальное решение. Оптимальный план имеет следующий вид:

$$x_1^*=40; x_2^*=0; x_3^*=0; x_4^*=80; x_5^*=0; x_6^*=0; x_7^*=120; \quad F^*=1240.$$

Таким образом, для того, чтобы прибыль от производства всей продукции была максимальна при имеющихся ограниченных ресурсах, необходимо выпустить 40 единиц продукции 1-го вида и 80 единиц продукции 4-го вида. Продукция 2-го и 3-го видов при этом не выпускается. Прибыль при этом равна 1240. Так как значения дополнительных переменных в оптимальном решении $x_5^*=0$ и $x_6^*=0$, то трудовые ресурсы и сырье полностью используются при производстве продукции. Финансовые ресурсы при производстве используются не полностью (имеется запас в 120 единиц).

3.2.2. Метод искусственного базиса

Для задачи линейной оптимизации, записанной в канонической форме, можно непосредственно указать опорный план, если в задаче имеются m базисных переменных, где m - число ограничений. Однако во многих задачах, записанных в канонической форме и имеющих опорные планы, не всегда есть m базисных переменных. Рассмотрим такую задачу:

$$\begin{aligned} F &= \sum_{j=1}^n c_j x_j \rightarrow \min, \\ \sum_{j=1}^n a_{ij} x_j &= b_i, i = \overline{1, m}, \\ x_j &\geq 0, j = \overline{1, n}. \end{aligned}$$

В данном случае для получения первоначального опорного плана в систему ограничений вводятся неотрицательные переменные, называемые *искусственными переменными*. Тогда система ограничений рассматриваемой задачи примет вид:

$$\sum_{j=1}^n a_{ij}x_j + x_{n+i} = b_i, i = \overline{1, m},$$

где x_{n+i} – искусственные переменные.

Так как введение искусственных переменных меняет множество решений задачи, то они вводятся и в выражение для целевой функции с очень большим коэффициентом $M > 0$:

$$F = \sum_{j=1}^n c_j x_j + M \sum_{j=n+1}^{n+m} x_j \rightarrow \min$$

Задача такого вида всегда называется *расширенной* по отношению к исходной. Тогда в процессе решения задачи минимизации искусственные переменные будут стремиться к нулю.

Первоначальное базисное решение будет иметь вид:

$$X = (0, \dots, 0, b_1, \dots, b_m)$$

Тогда целевая функция примет вид:

$$F = \sum_{j=1}^n c_j x_j + M \sum_{i=1}^m b_i \rightarrow \min$$

Оценки Δ_j в данном случае вычисляются следующим образом:

$$\Delta_j = \sum_{i=1}^m c_{Bi} a_{ij} - c_j = M \sum_{i=1}^m a_{ij} - c_j$$

Таким образом, оценки можно представить в виде: $\Delta_j = \alpha_j + M\beta_j$. Каждая j -я оценка состоит из двух частей, одна из которых зависит от M , а другая не зависит.

Исходные данные расширенной задачи заносят в симплекс-таблицу, которая содержит на одну строку больше, чем обычная. В $(m+2)$ -ю строку оценок таблицы записывают коэффициенты при M , а в $(m+1)$ -ю заносят слагаемые, не содержащие M .

Так как знак оценки Δ_j определяется знаком коэффициента, стоящего при M , ведущий столбец и базисную переменную выбирают по $(m+2)$ -й строке таблицы. Пересчет симплекс-таблицы при переходе от одного опорного плана к другому производят по общим правилам симплексного метода. Если в процессе решения задачи какая-либо искусственная переменная выводится из базиса, то соответствующий столбец симплекс-таблицы можно вычеркнуть и не пересчитывать (т. к. эту переменную не имеет смысла вводить ни в один из последующих базисов).

Итерационный процесс по $(m+2)$ -й строке симплекс-таблицы ведут до тех пор, пока:

1) либо все искусственные переменные не будут исключены из базиса;

2) либо не все искусственные переменные исключены, но $(m+2)$ -я строка не содержит больше положительных элементов.

В первом случае базис отвечает некоторому опорному плану исходной задачи и определение ее оптимального плана продолжают по $(m+1)$ -й строке.

Во втором случае, если элемент, стоящий в $(m+2)$ -й строке столбца B положителен, задача не имеет решения, если он равен нулю, то базис содержит по крайней мере один из векторов искусственного базиса.

Таким образом, процесс нахождения решения ЗЛП методом искусственного базиса включает следующие этапы:

1. Составление расширенной задачи.
2. Определение опорного плана расширенной задачи.
3. Вывод искусственных переменных из базиса с использованием симплекс-метода. В результате либо находят опорный план исходной задачи, либо устанавливают ее неразрешимость.
4. Определение оптимального плана исходной задачи с использованием симплекс-метода.

Пример 3.3. Решить задачу линейной оптимизации:

$$F=6x_1+x_2+x_3\rightarrow\min;$$

$$2x_1+4x_2+5x_3\leq 26;$$

$$2x_1+x_2\geq 8;$$

$$12x_1+3x_2+x_3\leq 60;$$

$$x_j\geq 0, \quad j=\overline{1,3}.$$

Приведем задачу к канонической форме.

$$F=6x_1+x_2+x_3\rightarrow\min;$$

$$2x_1+4x_2+5x_3+x_4=26;$$

$$2x_1+x_2-x_5=8;$$

$$12x_1+3x_2+x_3+x_6=60;$$

$$x_j\geq 0, \quad j=\overline{1,6}.$$

Составим расширенную задачу, вводя искусственную переменную x_7 в первое ограничение. Эту же переменную вводим в целевую функцию с большим коэффициентом M . Расширенная задача имеет вид:

$$F=6x_1+x_2+x_3+Mx_7\rightarrow\min;$$

$$2x_1+4x_2+5x_3+x_4=26;$$

$$2x_1+x_2-x_5+x_7=8;$$

$$12x_1+3x_2+x_3+x_6=60;$$

$$x_j\geq 0, \quad j=\overline{1,6}.$$

Решение задачи приведено в табл. 3.7.

В результате решения получен оптимальный план:

$$x_1^*=1; x_2^*=6; x_3^*=0; x_4^*=30; x_5^*=0; x_6^*=11; F(X^*)=12.$$

Таблица 3.7.

X_δ	c_δ	B	6	1	1	0	0	0	M
			x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_4	0	26	2	4	5	1	0	0	0
X_6	0	8	<u>2</u>	1	0	0	-1	0	1
X_7	M	60	12	3	1	0	0	1	0
0			-5	-1	-1	0	0	0	0
60			12	3	1	0	0	0	0
x_δ	c_δ	B	6	1	1	0	0	0	
			x_1	x_2	x_3	x_4	x_5	x_6	
x_4	0	18	0	<u>3</u>	5	1	1	0	
x_1	6	4	1	1/2	0	0	-1/2	0	
x_6	0	12	0	-3	1	0	6	1	
20			0	2	-1	0	-3	0	
x_δ	c_δ	b	6	1	1	0	0	0	
			x_1	x_2	x_3	x_4	x_5	x_6	
x_2	1	6	0	1	5/3	1/3	1/3	0	
x_1	6	1	1	0	-5/6	-1/6	-2/3	0	
x_4	0	30	0	0	6	1	7	1	
12			0	0	-10/3	-2/3	-11/3	0	

3.3. Этапы решения задач линейной оптимизации в EXCEL

Решение задач в среде EXCEL начинается с ввода условий задачи. Ввод условий задачи состоит из следующих основных шагов:

- создание формы для ввода условий задачи;
- ввод исходных данных;
- ввод зависимостей из математической модели;
- назначение целевой функции;
- ввод ограничений и граничных условий.

Последовательность работ рассмотрим на примере задачи составления производственного плана, рассмотренной в п.3.1. .

Математическая модель для решения данной задачи:

$$F=7x_1+3x_2+6x_3+12x_4 \rightarrow \max;$$

$$3x_1+x_2+2x_3+4x_4 \leq 440;$$

$$x_1+8x_2+6x_3+2x_4 \leq 200;$$

$$x_1+4x_2+7x_3+2x_4 \leq 320;$$

$$x_j \geq 0, \quad j=\overline{1,4}.$$

Форма для ввода условий задачи может иметь следующий вид:

	A	B	C	D	E	F	G	H
1				переменные				
2	имя	прод1	прод2	прод3	прод4			
3	значение							
4	нижн.гр.							
5	верхн.гр.							
6	коэф. в цф						ЦФ	направление
7				ограничения				
8	вид					левая часть	знак	правая часть
9	трудовые							
10	сырье							
11	финансы							

Рис. 3.1.

Весь текст на рис. 3.1 (и в дальнейшем) является комментарием и на решение задачи не влияет.

Далее необходимо ввести в сформированную форму исходные данные (коэффициенты в целевой функции и ограничениях, а также направление оптимизации целевой функции и знаки ограничений (рис. 3.2)

	А	В	С	Д	Е	Ф	Г	Н
2	имя	прод1	прод2	прод3	прод4			
3	значение							
4	нижн.гр.							
5	верхн.гр.							
6	коэф. в цф	7	3	6	12			max
7				ограничения				
8	вид					левая часть	знак	правая часть
9	трудовые	3	1	2	4		<=	440
10	сырье	1	8	6	2		<=	200
11	финансы	1	4	7	2		<=	320

Рис. 3.2

При этом ячейки В3:Е3 являются изменяемыми и в них будут заноситься значения переменных.

Ввод функциональных зависимостей для целевой функции и ограничений может осуществляться с использованием **Мастера функций**. Для этого необходимо активизировать требуемую ячейку (F6) и вызвать **Мастер функций**. В левой части появившегося диалогового окна нужно выбрать категорию функции **Математическая**, а в правом окне выделить функцию СУММПРОИЗВ и нажать клавишу **ОК**. Затем на экране отобразится диалоговое окно второго шага (рис. 3.3), где требуется ввести как первый (B\$3:E\$3), так и второй массивы (B6:E6). При вводе первого массива используются абсолютные ссылки на ячейки, при вводе второго - относительные, что в дальнейшем будет удобно при копировании формул. Во все окна адреса ячеек удобно вводить не с клавиатуры, а протаскивая мышью по соответствующим ячейкам.

СУММПРОИЗВ

Массив1 = {0;0;0;0}

Массив2 = {0;0;0;0}

Массив3 =

= 0

Возвращает сумму произведений соответствующих элементов массивов.

Массив2: массив1;массив2;... от 2 до 30 массивов, чьи компоненты нужно перемножить а затем сложить. Все массивы должны иметь одну и ту же размерность.

Значение: 0

OK Отмена

Рис.3.3.

Зависимости для левых частей ограничений вводятся аналогично. При этом необходимо лишь менять адреса ячеек. Для ускорения и удобства ввода можно скопировать содержимое ячейки F6 в ячейки F9, F10 и F11 (при этом все относительные ссылки изменятся автоматически).

Окончательная таблица с исходными данными представлена на рис. 3.4.

	A	B	C	D	E	F	G	H
1	.	переменные						
2	имя	прод1	прод2	прод3	прод4			
3	значение							
4	нижн.гр.							
5	верхн.гр.							
6	коэф. в цф	7	3	6	12	=СУММПРОИЗВ(B\$3:E\$3;B6:E6)	макс	
7				ограничения				
8	вид					левая часть	знак	пр.часть
9	трудовые	3	1	2	4	=СУММПРОИЗВ(B\$3:E\$3;B9:E9)	<=	440
10	сырье	1	8	6	2	=СУММПРОИЗВ(B\$3:E\$3;B10:E10)	<=	200
11	финансы	1	4	7	2	=СУММПРОИЗВ(B\$3:E\$3;B11:E11)	<=	320

Рис. 3.4

После окончания ввода исходных данных осуществляется вызов программы **Поиск решения**. Для этого необходимо выбрать в меню пункт **Сервис**, а в нем – **Поиск решения**, в результате чего на экране появится окно поиска решения (рис. 3.5).

Рис. 3.5

В окне **Установить целевую ячейку** требуется ввести имя ячейки для целевой функции (в данном случае F6). В качестве направления оптимизации выбирается максимизация. В окне **Изменяя ячейки** вводятся адреса ячеек, соответствующих варьируемому переменным задачи (\$B3:\$E3). Далее необходимо ввести ограничения. Для добавления ограничений выбирается пункт **Добавить**, после чего появляется окно добавления ограничений (рис. 3.6)

Рис. 3.6

Вводятся граничные условия для переменных (Прод1 - Прод4) ≥ 0 : $B3 \geq B4$, $C3 \geq C4$, $D3 \geq D4$, $E3 \geq E4$ (нулевые значения ячеек B4-E4 можно не устанавливать). Ограничения можно также ввести в виде $B3 \geq 0$, $C3 \geq 0$, $D3 \geq 0$, $E3 \geq 0$. Затем вводятся ограничения на ресурсы: $F9 \leq H9$, $F10 \leq H10$, $F11 \leq H11$. Ограничения вводятся последовательно. Сначала выбирается пункт **Добавить**, далее в появившихся диалоговых окнах вводится левая часть, знак и правая часть каждого ограничения. После ввода последнего ограничения и нажатия **ОК** произойдет возврат в окно **Поиск решения**.

Решение задачи производится сразу же после ввода данных, когда на экране находится диалоговое окно **Поиск решения**. Перед началом решения необходимо установить параметры решения, для чего в окне поиска решения выбрать команду **параметры**. Диалоговое окно параметров поиска решения представлено на рис. 3.7.

Параметры поиска решения		
Максимальное время:	100 секунд	ОК
Предельное число итераций:	100	Отмена
Относительная погрешность:	0,000001	Загрузить модель...
Допустимое отклонение:	5 %	Сохранить модель...
Сходимость:	0,001	Справка
☒ Линейная модель ☐ Автоматическое масштабирование		
☐ Неотрицательные значения ☐ Показывать результаты итераций		
Оценки	Разности	Метод поиска
☒ линейная	☒ прямые	☒ Ньютона
☐ квадратичная	☐ центральные	☐ сопряженных градиентов

Рис. 3.7

С помощью команд, находящихся в этом диалоговом окне, можно вводить условия для решения задач оптимизации всех клас-

сов. Рассмотрим наиболее важные команды, применяемые при решении конкретных задач.

Максимальное время

Служит для назначения времени в секундах, выделяемого на поиск решения задачи. В поле можно ввести время, не превышающее 32767 с (более 9 часов!). Значение 100, используемое по умолчанию, подходит для большинства задач.

Предельное число итераций

Служит для назначения числа итераций. Используемое по умолчанию число 100 подходит для решения большинства задач

Относительная погрешность

Используется для задания точности, с которой определяется соответствие ячейки целевому значению или приближение к указанным границам. Поле должно содержать число из интервала от 0 (нуля) до 1. Низкая точность соответствует введенному числу, содержащему меньшее количество десятичных знаков, чем число, используемое по умолчанию с например, 0,0001. Высокая точность увеличит время, которое требуется для того, чтобы сошелся процесс оптимизации.

Допустимое отклонение

Используется для задания допуска на отклонение от оптимального решения, если множество значений влияющей ячейки ограничено множеством целых чисел. При указании большего допуска поиск решения заканчивается быстрее.

Сходимость

Когда относительное изменение значения в целевой ячейке за последние пять итераций становится меньше числа, указанного в поле **Сходимость**, поиск прекращается. Сходимость применяется только к нелинейным задачам, условием служит дробь из интервала от 0 (нуля) до 1. Лучшую сходимость характеризует большее количество десятичных знаков. Лучшая сходимость требует больше времени на поиск оптимального решения.

Линейная модель

Используется для решения линейной задачи оптимизации или линейной аппроксимации нелинейной задачи.

Показывать результаты итераций

Используется для приостановки поиска решения для просмотра результатов отдельных итераций.

Автоматическое масштабирование

Служит для включения автоматической нормализации входных и выходных значений, качественно различающихся по величине, например, максимизация прибыли в процентах по отношению к вложениям в миллионах рублей.

Значения не отрицательны

Позволяет установить нулевую нижнюю границу для тех влияющих ячеек, для которых она не была указана в поле **Ограничение** диалогового окна **Добавить ограничение**.

Остальные команды будут рассмотрены позже при изучении соответствующих классов задач.

Для решения задачи линейного программирования необходимо установить флажок **Линейная модель** в окне **Параметры поиска решения**, что обеспечит использование симплексного метода. Далее после выбора **ОК** произойдет возврат в окно **Поиск решения**. Запуск процесса решения задачи осуществляется командой **Выполнить**.

Решение задачи занимает несколько секунд, после чего на экране появляется диалоговое окно **Результаты поиска решения**. При решении рассматриваемой задачи производственного планирования в окне появится сообщение о том, что решение найдено. В данном окне предоставляется возможность или сохранить найденное решение во влияющих ячейках модели, или восстановить исходные данные. Для этого необходимо отметить нужный пункт.

Результирующие значения всех переменных записываются в ячейки В3-Е3 исходной задачи. Соответствующее значение целевой

функции заносится в ячейку F6, а значения левых частей ограничений – в ячейки F9-F11. Таблица с результатами решения задачи представлена на рис. 3.8.

	A	B	C	D	E	F	G	H
1			переменные					
2	имя	прод1	прод2	прод3	прод4			
3	значение	40	0	0	80			
4	нижн.гр.							
5	верхн.гр.							
6	коэф. в цф	7	3	6	12	1240	макс	
7			ограничения					
8	вид					левая часть	знак	правая часть
9	трудовые	3	1	2	4	440	<=	440
10	сырье	1	8	6	2	200	<=	200
11	финансы	1	4	7	2	200	<=	320

Рис. 3.8

Рис. 3.8 показывает, что в оптимальном решении

$$\text{Прод1} = B3 = 40;$$

$$\text{Прод2} = C3 = 0;$$

$$\text{Прод3} = D3 = 0;$$

$$\text{Прод4} = E3 = 80.$$

При этом максимальная прибыль будет составлять $F6 = 1240$, а количество использованных ресурсов равно:

$$\text{трудовых} = F9 = 440, \text{сырья} = F10 = 200, \text{финансов} = F11 = 200.$$

Таково оптимальное решение рассматриваемой задачи производственного планирования. Однако решение задачи находится не всегда. Если условия задачи несовместны, на экране появится надпись - **Поиск не может найти подходящего решения**. Если целевая функция не ограничена, то в окне результатов поиска решения появится сообщение: **Значения целевой функции не сходятся**.

Кроме этого, из окна **Результаты поиска решения** возможно создание отчётов трех типов: по результатам, устойчивости и пределам, каждый из которых записывается на отдельном листе. Отчеты используются при анализе полученного оптимального решения.

Отчет по результатам (рис. 3.9) состоит из трех таблиц.

В таблице 1 (**целевая ячейка**) приведены сведения о целевой функции, в столбце "Исходно" - указывается значение целевой функции до вычислений, "Результат" - указывается значение целевой функции после вычислений.

Таблица 2 (**изменяемые ячейки**) содержит сведения о значениях искомых переменных, до и после решения задачи.

Таблица 3 (**ограничения**) показывает результаты оптимального решения для ограничений и для граничных условий. Здесь в графе "Формула" приведены зависимости, которые были введены в диалоговом окне "Поиск решения"); в графе "Разница" показано количество неиспользованного ресурса. Если ресурс используется полностью, то в графе "Состояние" указывается *связанное*; при неполном использовании ресурса в этой графе указывается *не связан*.

Целевая ячейка (Максимум)

Ячейка	Имя	Исходно	Результат
\$F\$6	коэф. в цф	1240	1240

Изменяемые ячейки

Ячейка	Имя	Исходно	Результат
\$B\$3	значение прод1	40	40
\$C\$3	значение прод2	0	0
\$D\$3	значение прод3	0	0
\$E\$3	значение прод4	80	80

Ограничения

Ячейка	Имя	Значение	формула	Статус	Разница
\$F\$10	сырье левая часть	200	\$F\$10<=\$H\$10	связанное	0
\$F\$11	финансы левая часть	200	\$F\$11<=\$H\$11	не связан.	120
\$F\$9	трудовые левая часть	440	\$F\$9<=\$H\$9	связанное	0
\$E\$3	значение прод4	80	\$E\$3>=\$E\$4	не связан.	80
\$B\$3	значение прод1	40	\$B\$3>=\$B\$4	не связан.	40
\$C\$3	значение прод2	0	\$C\$3>=\$C\$4	связанное	0
\$D\$3	значение прод3	0	\$D\$3>=\$D\$4	связанное	0

Рис. 3.9

Отчет по устойчивости (рис. 3.10) состоит из двух таблиц.

В таблице 1 (**изменяемые ячейки**) приводятся следующие значения для переменных:

- результат решения задачи;
- редуцированная стоимость, т.е. дополнительные двойственные переменные v_j , которые показывают, насколько изменяется целевая функция при принудительном включении единицы этой продукции в оптимальное решение;
- коэффициенты целевой функции;
- предельные значения приращения коэффициентов Δc_j целевой функции при которых сохраняется набор переменных, входящих в оптимальное решение.

В таблице 2 (**ограничения**) приводятся аналогичные значения для ограничений:

- величины использованных ресурсов;
- теневые цены, которые показывают, как изменится целевая функция при изменении ресурсов на единицу;
- значения приращения ресурсов Δb_i , при которых сохраняется оптимальный набор переменных, входящих в решение.

Изменяемые ячейки						
Ячейка	Имя	Результ. значение	Нормир. стоимость	Целевой Коэффициент	Допустимое Увеличение	Допустимое Уменьшение
\$B\$3	значение прод1	40	0	7	2	1
\$C\$3	значение прод2	0	-30	2,999999999	30	1E+30
\$D\$3	значение прод3	0	-20	6,000000001	20	1E+30
\$E\$3	значение прод4	80	0	12	2	2,5

Ограничения						
Ячейка	Имя	Результ. значение	Теневая Цена	Ограничение Правая часть	Допустимое Увеличение	Допустимое Уменьшение
\$F\$10	сырье левая часть	200	4	200	20	53,33333333
\$F\$11	финансы левая часть	200	0	320	1E+30	120
\$F\$9	трудовые левая часть	440	1	440	160	40

Рис. 3.10

Отчет по пределам (рис. 3.11) показывает, в каких пределах может изменяться выпуск продукции, вошедшей в оптимальное решение, при сохранении структуры оптимального решения:

- рассматриваются значения x_j в оптимальном решении;
- рассматриваются нижние пределы изменения значений x_j .

Кроме этого, в отчете указаны значения целевой функции при выпуске данного типа продукции на нижнем пределе. Далее приводятся верхние пределы изменения x_j и значения целевой функции при выпуске продукции, вошедшей в оптимальное решение на верхних пределах.

Целевое		
Ячейка	Имя	значение
\$F\$6	коэф. в цф	1240

Изменяемое		
Ячейка	Имя	значение
\$B\$3	значение прод1	40
\$C\$3	значение прод2	0
\$D\$3	значение прод3	0
\$E\$3	значение прод4	80

Нижний предел	Целевой результат
0	960
0	1240
0	1240
0	280

Верхний предел	Целевой результат
40	1240
0	1240
0	1240
80	1240

Рис. 3.11

Важным фактором, помогающим принять оптимальное решение, является наглядное представление полученного результата. Результат решения задачи, рассмотренной выше, примем в качестве исходных данных при рассмотрении вопросов построения диаграмм.

Для построения диаграмм необходимо выделить те ячейки, значения которых должны быть представлены на диаграмме (B3:E3). Затем вызывается **Мастер диаграмм** (рис. 3.12). В появившемся диалоговом окне выбирается нужный тип диаграммы. Выберем, например, объемный вариант гистограммы.

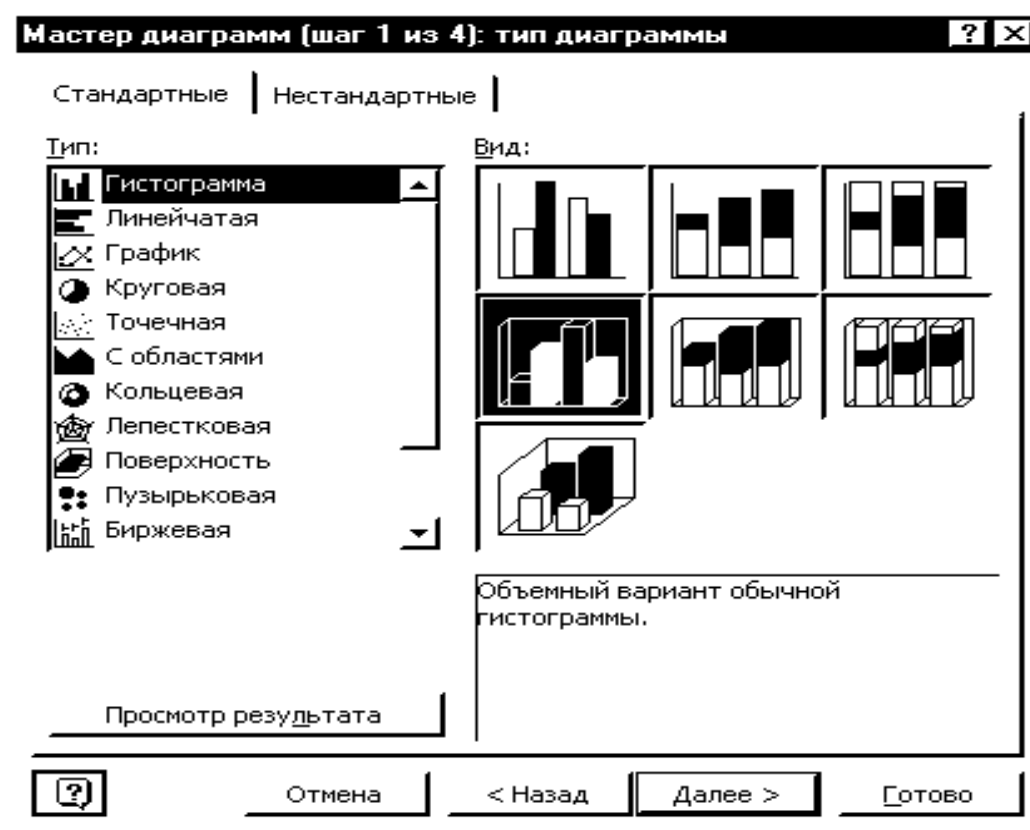


Рис. 3.12

После выбора типа диаграммы осуществляется переход к следующему этапу построения диаграммы. В диалоговом окне **Исходные данные** (рис. 3.13) сначала необходимо ввести диапазон данных, которые отображаются на диаграмме (в нашем примере ячейки В3:Е3), а затем, выбрав команду **Ряд**, ввести названия рядов (например, “Количество продукции”), а также подписи под осью X (Прод1, Прод2, Прод3, Прод4). Для ввода подписей можно выделить соответствующие ячейки таблицы (В2:Е2) или ввести подписи непосредственно в диалоговом окне через двоеточие.

После окончания работы с окном **Исходные данные** необходимо выбрать пункт **Далее** и перейти в диалоговое окно **Параметры диаграммы** (рис. 3.14). В данном окне, если необходимо, вводится заголовок диаграммы, размещается легенда, подписываются оси, определяются линии сетки. Все вводимые изменения отображаются на диаграмме. Затем осуществляется переход в диалоговое

окно **Размещение диаграммы** и выбирается тип размещения: на отдельном рабочем листе или на имеющемся рабочем листе. После выбора пункта **Готово** диаграмма размещается на указанном листе. Построенная гистограмма для рассматриваемой задачи представлена на рис. 3.15.

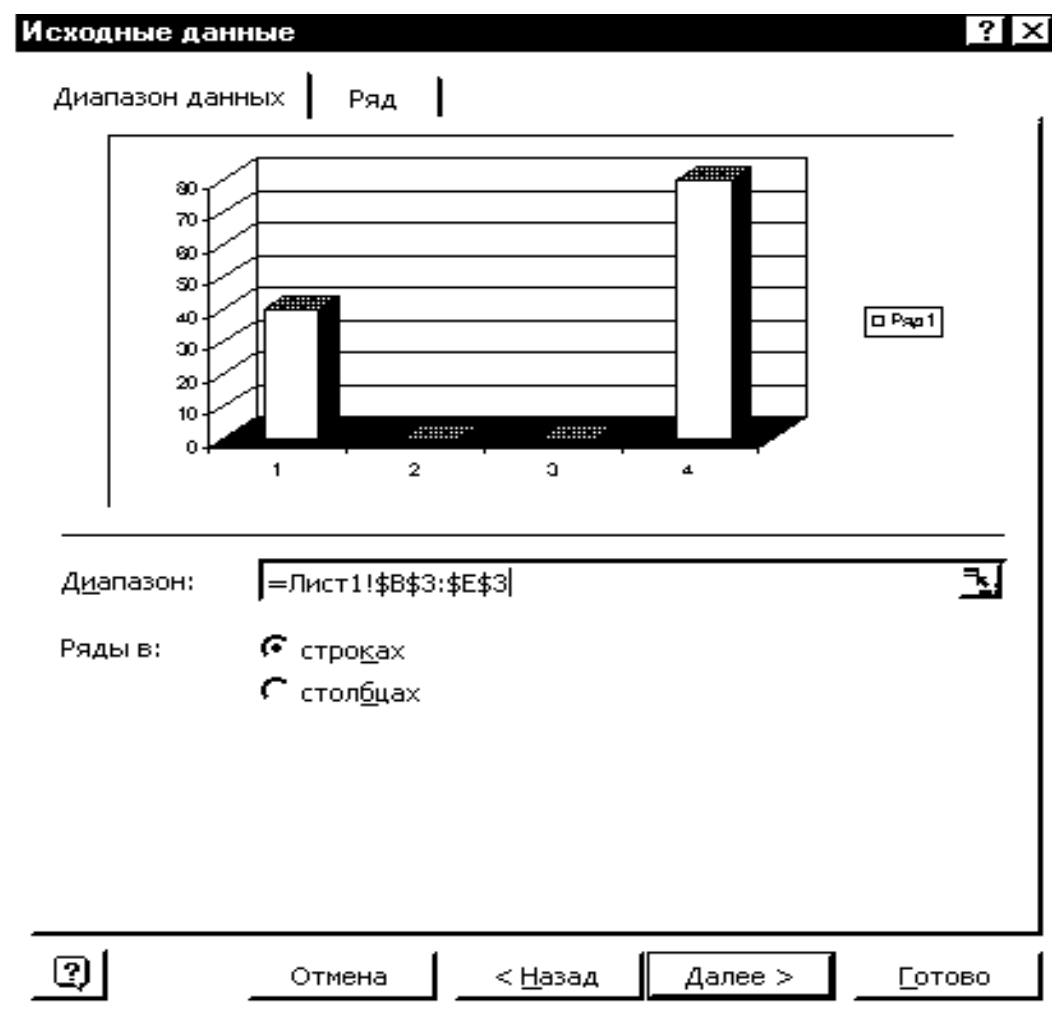


Рис. 3.13

Возможно перемещение данной диаграммы, изменение её размеров, перестройка параметров. Для этого необходимо активизировать нужную диаграмму с помощью мыши и в режиме форматирования диаграммы выполнить необходимые действия.

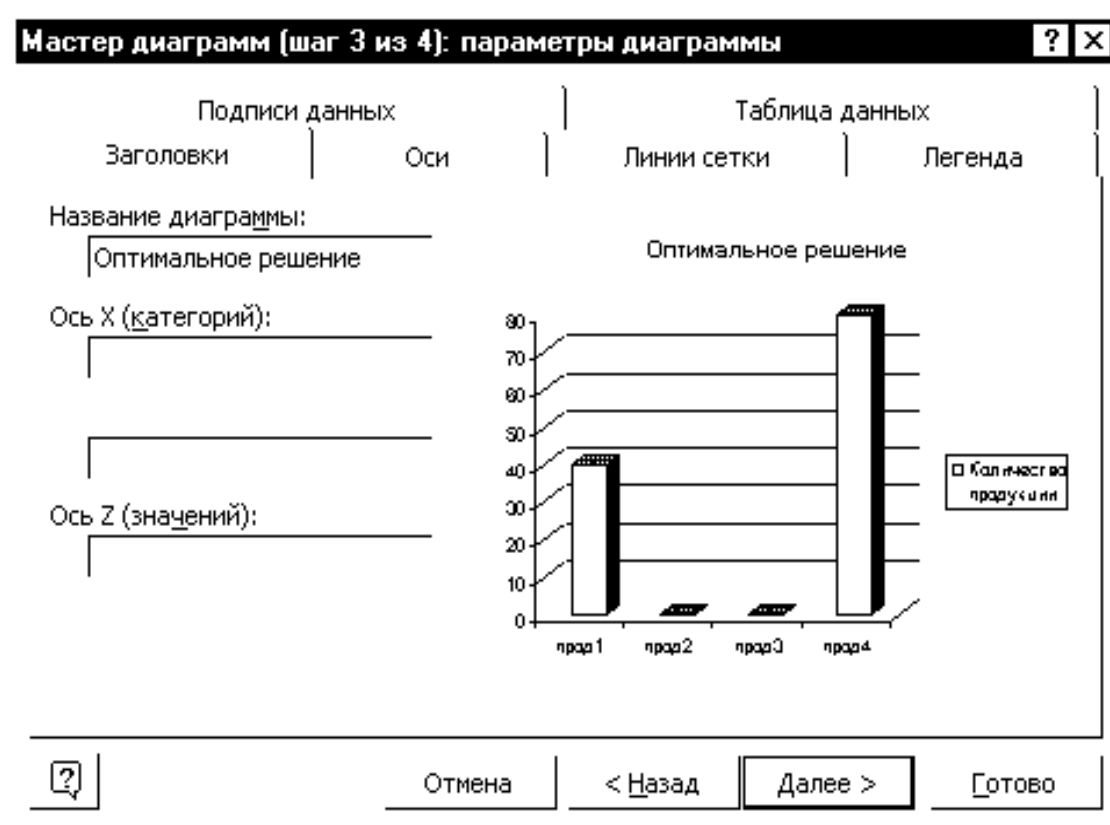


Рис. 3.14



Рис. 3.15

Параметрический анализ

Под параметрическим анализом будем понимать решение задачи оптимизации при различных значениях того параметра, который ограничивает улучшение целевой функции.

Параметрический анализ будем выполнять для рассматриваемой задачи производственного планирования, решая её при различных значениях имеющегося сырья. Составим таблицу вариантов:

Вариант	1	2	3	4	5
Сырье	100	150	200	250	300

Для выполнения параметрических расчетов в таблице с результатами решения задачи (рис. 3.8) необходимо удалить результат решения, находящийся в ячейках В3:Е3. Далее решить задачу для первого варианта по описанной выше схеме, предварительно введя в ячейку Н10 значение 100. После решения задачи в диалоговом окне **Результаты поиска решения** выбирается команда **Сохранить сценарий**. При этом на экране появится диалоговое окно **Сохранение сценария** (рис. 3.16). В появившемся диалоговом окне необходимо ввести имя сценария “Сырье=100” и подтвердить ввод выбором кнопки **ОК**.

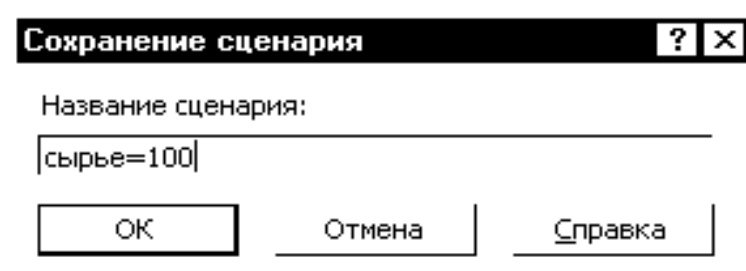


Рис. 3.16

Результат решения задачи для данного варианта представлен на рис. 3.17.

	A	B	C	D	E	F	G	H
1				переменные				
2	имя	прод1	прод2	прод3	прод4			
3	значение	100	0	0	0			
4	нижн.гр.							
5	верхн.гр.							
6	коэф. в цф	7	3	6	12	700	макс	
7				ограничения				
8	вид					левая часть	знак	правая часть
9	трудовые	3	1	2	4	300	<=	440
10	сырье	1	8	6	2	100	<=	100
11	финансы	1	4	7	2	100	<=	320

Рис. 3.17

Далее необходимо аналогично решить задачу для всех оставшихся вариантов, последовательно вводя в ячейку H10 значения 150, 200, 250, 300. При этом нужно сохранять каждый сценарий и вводить его имя, соответствующее текущему значению сырья.

Для представления результатов решения вызывается пункт меню **Данные, Анализ “Что-Если”** и в появившемся диалоговом окне **Диспетчер сценариев** (рис. 3.18) выбирается пункт **Отчет**.

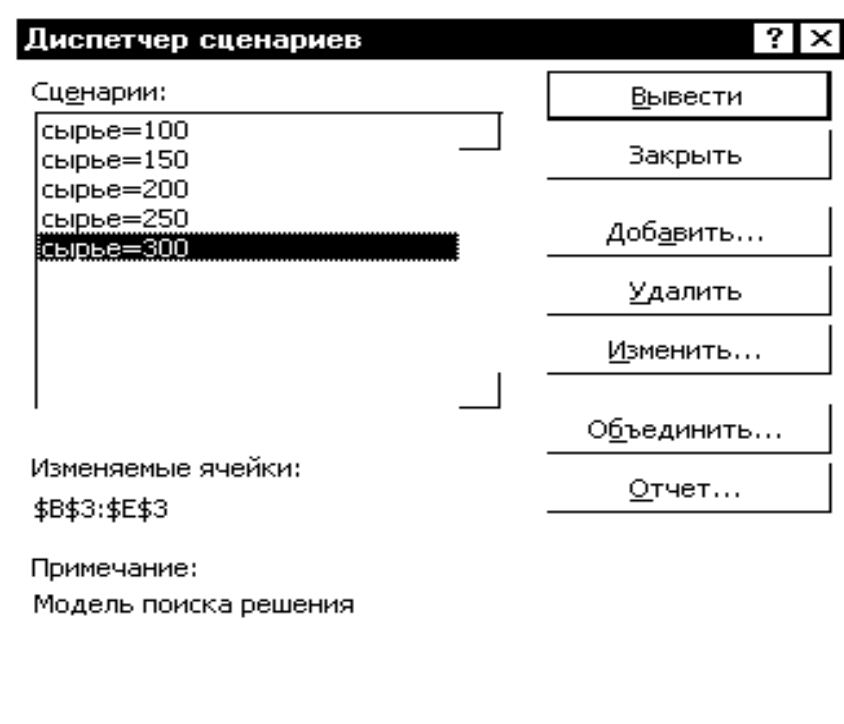


Рис. 3.18

При выборе в окне **Отчет по сценарию** (рис. 3.19) типа отчета **Структура** создается итоговый сценарий (рис.3.20), который размещается на отдельном листе с названием “Структура сценария”.

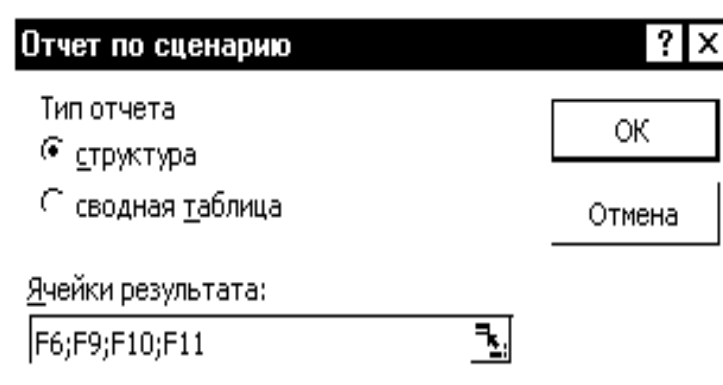


Рис. 3.19

	A	B	C	D	E	F	G	H
1	Структура сценария							
2	Текущие значения:			сырье=100	сырье=150	сырье=200	сырье=250	сырье=300
4	Изменяемые:							
5		\$B\$3	0	100	140	40	0	0
6		\$C\$3	0	0	0	0	0	0
7		\$D\$3	0	0	0	0	0	0
8		\$E\$3	110	0	5	80	110	110
9	Результат:							
10		\$F\$6	1320	700	1040	1240	1320	1320
11		\$F\$9	440	300	440	440	440	440
12		\$F\$10	220	100	150	200	220	220
13		\$F\$11	220	100	150	200	220	220
14	Примечания: столбец "Текущие значения" представляет значения изменяемых ячеек в							
15	момент создания Итогового отчета по Сценарию. Изменяемые ячейки для каждого							
16	сценария выделены серым цветом.							

Рис. 3.20

Для удобства дальнейшей работы полученный сценарий можно отредактировать и представить следующем виде:

	A	B	C	D	E	F
1	ИТОГОВЫЙ СЦЕНАРИЙ					
2		сырье=100	сырье=150	сырье=200	сырье=250	сырье=300
4	Прод1	100	140	40	0	0
5	Прод2	0	0	0	0	0
6	Прод3	0	0	0	0	0
7	Прод4	0	5	80	110	110
8	Прибыль	700	1040	1240	1320	1320
9	Трудовые	300	440	440	440	440
10	сырье	100	150	200	220	220
11	финансы	100	150	200	220	220

Рис. 3.21

Для наглядного представления результатов построим гистограммы:



Рис. 3.22

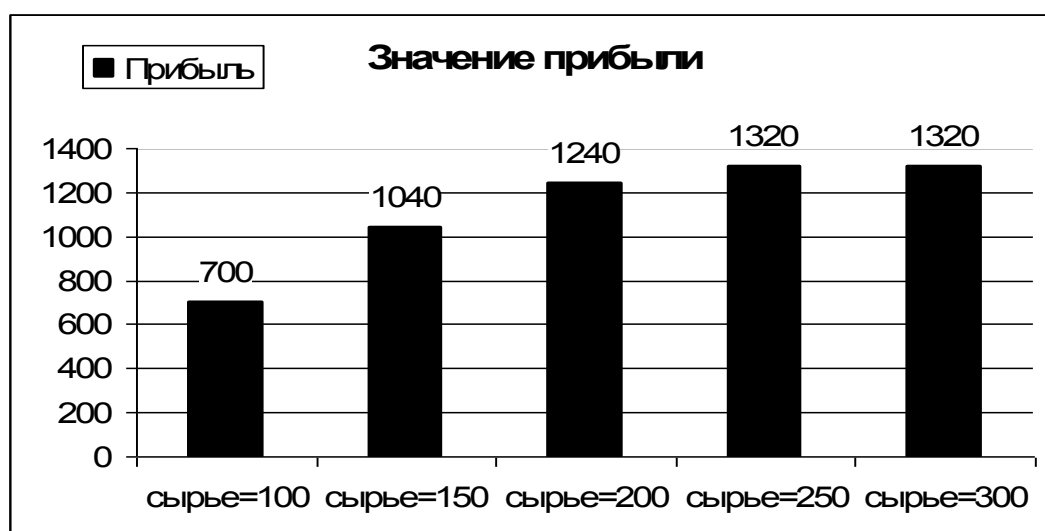


Рис. 3.23.

3.4. Лабораторная работа №2

РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОЙ ОПТИМИЗАЦИИ СРЕДСТВАМИ EXCEL.

Цель работы: Получение практических навыков формирования математических моделей прикладных задач линейной оптимизации и их решения средствами EXCEL.

Лабораторные задания.

1. По п.3.1 - 3.3 пособия изучить задачи линейного программирования и методы их решения.

2. Выбрать задачу в соответствии с вариантом, составить ее математическую модель и решить с использованием средств EXCEL. Сформировать отчеты по устойчивости, результатам и пределам. Для визуализации результатов решения построить диаграммы. Проанализировать полученные результаты.

3. Провести параметрический анализ решая задачу при пяти различных значениях ресурса оборудования первого типа (варианты 1-3), второго типа (варианты 4-7) и третьего типа (варианты 8-10). Значения менять как в большую, так и в меньшую сторону.

4. Решить задачу **вручную** с использованием симплекс-метода и графического метода.

5. Сравнить результаты, полученные при выполнении лабораторных заданий 1 и 2.

Задания

Для производства двух видов изделий А и В используются три типа технологического оборудования. Для производства одного изделия А оборудование первого типа используется в течение a_1 часов, оборудование второго типа – a_2 часов, оборудование третьего типа – a_3 часов. Для производства одного изделия В оборудование первого типа используется в течение b_1 часов, оборудование второго типа –

b_2 часов, оборудование третьего типа – b_3 часов. На изготовление всех изделий предприятие может использовать оборудование первого типа не более чем t_1 часов, оборудование второго типа – не более t_2 часов, оборудование третьего типа – не более t_3 часов. Прибыль от реализации одного готового изделия А составляет α денежных единиц, а изделия В - β денежных единиц. Составить план производства изделий А и В, обеспечивающий максимальную прибыль от их реализации.

Вариант	a_1	a_2	a_3	b_1	b_2	b_3	t_1	t_2	t_3	α	β
1	1	1	3	1	2	1	20	36	40	2	5
2	2	1	3	2	2	1	40	34	46	1	2
3	3	1	3	3	2	1	60	32	50	1	3
4	1	1	3	1	2	1	24	40	52	2	4
5	2	1	3	2	2	1	48	38	56	2	6
6	3	2	1	1	2	2	40	40	36	2	1
7	3	1	1	1	2	1	46	34	20	5	2
8	1	3	3	2	3	1	32	60	50	4	2
9	1	3	1	2	1	1	40	52	24	6	2
10	3	2	1	1	2	2	56	48	38	3	1

Отчет по лабораторной работе должен содержать :

1. Постановку задачи.
2. Математическую оптимизационную модель.
3. Краткое описание основных этапов решения в среде EXCEL.
4. Результат решения задачи (итоговую таблицу в EXCEL).
5. Отчеты по результатам, устойчивости, пределам.
6. Построенные диаграммы.
7. Результаты параметрического анализа с соответствующими выводами.
8. Решение задачи вручную с использованием симплекс-метода и графического метода.
10. Выводы.

4. ТРАНСПОРТНЫЕ ЗАДАЧИ ЛИНЕЙНОЙ ОПТИМИЗАЦИИ

4.1. Особенности задач транспортного типа

Одним из наиболее значимых классов задач линейной оптимизации являются задачи транспортного типа. Рассмотрим постановку транспортной задачи.

Пусть имеется m пунктов отправления $A_1, A_2, \dots, A_m$, на которых сосредоточен однородный груз в количествах $a_1, a_2, \dots, a_m$ единиц, и n пунктов назначения $B_1, B_2, \dots, B_n$, потребности которых в данном грузе равны $b_1, b_2, \dots, b_n$. Стоимость перевозки единицы груза из пункта A_i в пункт B_j (тариф) составляет c_{ij} единиц. Требуется составить план перевозок, позволяющий вывезти все грузы из пунктов отправления, удовлетворить все потребности в грузах в пунктах назначения и имеющий минимальную стоимость.

Обозначим через x_{ij} количество единиц груза, перевозимое из i -го пункта отправления (A_i) в j -й пункт назначения (B_j).

Математическая модель задачи имеет следующий вид:

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$$
$$\sum_{j=1}^n x_{ij} = a_i, \quad i = \overline{1, m} \quad (4.1)$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = \overline{1, n} \quad (4.2)$$

$$x_{ij} \geq 0$$

Ограничения (4.1) означают, что весь груз должен быть полностью вывезен из пунктов отправления.

Ограничения (4.2) означают, что все потребности в грузах должны быть удовлетворены.

Модель транспортной задачи называется *закрытой*, если потребность в грузе в пунктах назначения равна общему запасу груза в пунктах отправления: $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$. В противном случае модель

транспортной задачи называется *открытой*.

Для разрешимости транспортной задачи необходимо и достаточно, чтобы ее модель была закрытой. Приведение открытых моделей к закрытым осуществляется следующим образом:

1. Если запасы превышают потребности, т. е. $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$, то

вводится фиктивный (n+1)-й пункт назначения с потребностью

$b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$, а соответствующие тарифы считаются равны-

ми нулю: $c_{in+1} = 0 \ (i = \overline{1, m})$.

2. Аналогично при $\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$ вводится фиктивный (m+1)-й

пункт отправления с запасом груза $a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$ и соответ-

ствующие тарифы полагаются равными нулю: $c_{m+1j} = 0 \ (j = \overline{1, n})$. В результате получают закрытую модель транспортной задачи.

Если опорный план содержит n+m-1 отличных от нуля компонент, то он является *невыврожденным*, если меньше – *вырожденным*.

Для решения транспортных задач можно использовать симплекс-метод. Однако более рациональным является использование

специальных методов, учитывающих особенности задач данного класса.

4.2. Решение транспортных задач методом потенциалов

Решение транспортной задачи состоит из двух этапов:

- определение первоначального плана перевозки грузов;
- улучшение первоначального плана и построение оптимального плана.

Как и в симплекс-методе, решение транспортной задачи начинается с определения ее первоначального опорного плана. При нахождении опорного плана транспортной задачи на каждом шаге в таблице заполняют одну клетку, которую называют *занятой*. При этом либо полностью удовлетворяется потребность в грузе у одного из потребителей, либо обеспечивается полный вывоз груза из одного пункта отправления. В первом случае временно исключают из рассмотрения столбец, содержащий заполненную на данном шаге клетку, во втором случае – строку. Всего выполняется $n+m-1$ шаг при построении опорного плана.

Основными методами определения начального опорного плана транспортной задачи являются *метод северо-западного угла* и *метод минимального элемента*.

В *методе северо-западного угла* заполнение клеток матрицы планирования начинается с верхней левой клетки таблицы для неизвестного x_{11} (“северо-западный угол”) и заканчивается клеткой для неизвестного x_{mn} , т. е. идет по диагонали таблицы. Метод северо-западного угла может оказаться далеким от оптимального, так как не учитывает стоимость перевозки единицы груза. Опорный план, более близкий к оптимальному, дает метод минимального элемента.

В *методе минимального элемента* на каждом шаге выбирают для заполнения клетку, соответствующую минимальному тарифу c_{ij} (если таких клеток несколько, то выбирается любая из них).

При определении первоначального опорного плана необходимо следить за тем, чтобы полученный план оказался невырожденным, т. е. число заполненных клеток таблицы было строго равно $n+m-1$. Обозначим число заполненных клеток через k . Если $k < n+m-1$, то в $n+m-1-k$ незаполненных клеток помещаются фиктивные нулевые перевозки, и эти клетки считаются занятыми. При этом нулевые перевозки помещаются в те свободные клетки, которым соответствует наименьший тариф.

После построения первоначального опорного плана транспортной задачи осуществляется определение его оптимального плана. Наиболее широко при этом используется *метод потенциалов*.

Основные шаги метода потенциалов:

1. Построение начального опорного плана (метод северо-западного угла или метод минимального элемента).

2. Определение потенциалов пунктов отправления (u_i) и назначения (v_j). Их находят из системы уравнений $u_i + v_j = c_{ij}$, которая составляется для занятых клеток таблицы. Система содержит $n+m$ неизвестных и $n+m-1$ уравнение (т. к. число занятых клеток $n+m-1$). Поэтому одно из неизвестных приравнивается произвольному числу (например, $u_1 = 0$) и находятся значения остальных потенциалов.

3. Проверка оптимальности плана. Для этого проверяется выполнение условий $u_i + v_j \leq c_{ij}$ для свободных клеток таблицы. Вычисляются Δ_{ij} (оценки свободных переменных для исходного плана): $\Delta_{ij} = u_i + v_j - c_{ij}$. Если все $\Delta_{ij} \leq 0$, то найденный план оптимален. В противном случае, если для некоторой свободной клетки $\Delta_{ij} > 0$, то осуществляется переход к шагу 4.

4. Выбор клетки для заполнения на следующем шаге. Для этого просматриваются все свободные клетки, для которых $\Delta_{ij} > 0$, и для заполнения выбирается та, которой соответствует максимальное значение Δ_{ij} . Если таких клеток несколько, то для заполнения выбирается клетка с наименьшим тарифом.

5. Построение цикла пересчета для выбранной свободной клетки таблицы. *Циклом* в таблице условий транспортной задачи называется замкнутая ломаная, начинающаяся в выбранной свободной клетке таблицы, вершины которой (кроме начальной) расположены в занятых клетках таблицы, а звенья вдоль строк и столбцов. При правильном построении опорного плана для каждой свободной клетки можно построить только один цикл. Если ломаная линия, образующая цикл, пересекается, то точки самопересечения не являются вершинами. После построения цикла его вершины последовательно отмечаются знаками “–” или “+”, причем выбранной свободной клетке приписывается знак “+”.

6. Определение величины перераспределения груза d . Величину перераспределения груза определяют по следующему правилу: $d = \min(x_{ij})$, где x_{ij} – перевозки, стоящие в вершинах цикла, отмеченных знаком “–”.

7. Сдвиг по циклу пересчета (перераспределение грузов). В выбранную для заполнения свободную клетку, помеченную знаком “+”, заносится значение d . Одновременно это число прибавляется к значениям перевозок, стоящих в клетках со знаком “+” и вычитаются из значений перевозок, стоящих в клетках, отмеченных знаком “–”. Клетка, которая ранее была свободной, становится занятой, а минусовая клетка, в которой стояло значение d , становится свободной.

Замечание. При сдвиге по циклу пересчета число занятых клеток должно оставаться неизменным, т. е. $n+m-1$. Если минимальная перевозка d соответствует нескольким минусовым клеткам, то освобождают лишь одну из них, а в остальные записывают нулевые перевозки и считают занятыми.

8. Переход к шагу 2.

Пример 4.1. На 3 складах сосредоточен однородный груз в количествах 80, 100, и 70 единиц. Данный груз необходимо доставить 4 потребителям, потребности которых равны соответственно 80, 50, 50, 70 единиц. Матрица тарифов имеет вид:

$$\begin{pmatrix} 4 & 2 & 3 & 1 \\ 6 & 3 & 5 & 6 \\ 3 & 2 & 6 & 3 \end{pmatrix}$$

Необходимо определить оптимальный план перевозок.

Решение. Составим матрицу планирования. Начальный план перевозок определим методом минимального элемента. При этом сначала помещаем 70 единиц груза в клетку с индексом 14, имеющую минимальный тариф, затем 10 единиц в клетку 12, 40 единиц в клетку 32, 30 единиц в клетку 31, 50 единиц в клетку 23, 50 единиц в клетку 21.

Пункты отправления	Пункты назначения				Запасы
	B ₁	B ₂	B ₃	B ₄	
A ₁	4 70	2 10	3	1 70	80
A ₂	6 50	3	5 50	6	100
A ₃	3 30	2 40	6	3	70
Потребности	80	50	50	70	

Затем определяем потенциалы пунктов отправления и потенциалы пунктов назначения. Для этого составляем уравнения для занятых клеток таблицы:

$$U_1 + V_2 = 2$$

$$U_1 + V_4 = 1$$

$$U_2 + V_1 = 6$$

$$U_2 + V_3 = 5$$

$$U_3 + V_1 = 3$$

$$U_3 + V_2 = 2$$

Для решения системы положим $U_1 = 0$ и последовательно найдем значения остальных потенциалов. В результате получим: $U_1 = 0, U_2 = 3, U_3 = 0, V_1 = 3, V_2 = 2, V_3 = 2, V_4 = 1$.

Вычислим оценки Δ_{ij} для всех свободных клеток таблицы

$$\Delta_{11} = 0 + 3 - 4 = -1$$

$$\Delta_{13} = 0 + 2 - 3 = -1$$

$$\Delta_{22} = 3 + 2 - 3 = 2$$

$$\Delta_{24} = 3 + 1 - 6 = -2$$

$$\Delta_{33} = 0 + 2 - 6 = -4$$

$$\Delta_{34} = 0 + 1 - 3 = -2$$

Выбираем максимальную положительную оценку. Это оценка $\Delta_{22} = 2$. Следовательно, на следующем шаге будет заполняться клетка с индексом 22. Составим цикл пересчета для данной клетки и пометим его вершины последовательно знаками «+» и «-»

Пункты от- правления	Пункты назначения				Запасы
	B_1	B_2	B_3	B_4	
A_1	4 10	2	3	1 70	80
A_2	- 6 50	+ 3	5 50	6	100
A_3	+ 3 30	- 2 40	6	3	70
Потребности	80	50	50	70	

Определим величину перераспределения груза как минимальную из величин, стоящих в минусовых клетках. Это 40 единиц. Прибавляем данную величину к величинам перевозок, стоящим в

клетках со знаком «+» и вычитаем ее из величин перевозок, стоящих в клетках со знаком «-»»

Пункты отправления	Пункты назначения				Запасы
	B ₁	B ₂	B ₃	B ₄	
A ₁	4	2 10	3	1 70	80
A ₂	6 10	3 40	5 50	6	100
A ₃	3 70	2	6	3	70
Потребности	80	50	50	70	

Составляем систему уравнений для определения потенциалов и последовательно находим из нее значения всех потенциалов.

$$U_1 + V_2 = 2$$

$$U_1 + V_4 = 1$$

$$U_2 + V_1 = 6$$

$$U_2 + V_2 = 3$$

$$U_2 + V_3 = 5$$

$$U_3 + V_1 = 3$$

Тогда $U_1 = 0$, $U_2 = 1$, $U_3 = -2$, $V_1 = 5$, $V_2 = 2$, $V_3 = 4$, $V_4 = 1$. Пересчитываем оценки

$$\Delta_{11} = 0 + 5 - 4 = 1$$

$$\Delta_{13} = 0 + 4 - 3 = 1$$

$$\Delta_{24} = 1 + 1 - 6 = -4$$

$$\Delta_{32} = -2 + 2 - 2 = -2$$

$$\Delta_{33} = -2 + 4 - 6 = -4$$

$$\Delta_{34} = -2 + 1 - 3 = -4$$

Имеем две максимальные положительные оценки $\Delta_{11} = \Delta_{13} = 1$. Для заполнения на следующем шаге целесообразно выбрать клетку 13, т. к. ей соответствует меньший тариф. Строим цикл пересчета для клетки 13

Пункты отправления	Пункты назначения				Запасы
	B ₁	B ₂	B ₃	B ₄	
A ₁	4 10	- 2 10	+ 3 70	1	80
A ₂	6 10	+ 3 40	- 5 50	6	100
A ₃	3 70	2	6	3	70
Потребности	80	50	50	70	

Величина перераспределения груза равна 10. Перераспределяем груз

Пункты отправления	Пункты назначения				Запасы
	B ₁	B ₂	B ₃	B ₄	
A ₁	4	2	3 10	1 70	80
A ₂	6 10	3 50	5 40	6	100
A ₃	3 70	2	6	3	70
Потребности	80	50	50	70	

Находим потенциалы пунктов отправления и назначения:

$$U_1 + V_3 = 3$$

$$U_1 + V_4 = 1$$

$$U_2 + V_1 = 6$$

$$U_2 + V_2 = 3$$

$$U_2 + V_3 = 5$$

$$U_3 + V_1 = 3,$$

откуда $U_1 = 0, U_2 = 2, U_3 = -1, V_1 = 4, V_2 = 1, V_3 = 3, V_4 = 1$.

Пересчитываем оценки:

$$\Delta_{11} = 0 + 4 - 4 = 0$$

$$\Delta_{12} = 0 + 1 - 2 = -1$$

$$\Delta_{24} = 2 + 1 - 6 = -3$$

$$\Delta_{32} = -1 + 1 - 2 = -2$$

$$\Delta_{33} = -1 + 3 - 6 = -4$$

$$\Delta_{34} = -1 + 1 - 3 = -3$$

Все оценки неотрицательны. Следовательно, полученный план перевозок является оптимальным. Общая стоимость перевозки груза при этом составит: $10 \cdot 3 + 70 + 10 \cdot 6 + 50 \cdot 3 + 40 \cdot 5 + 70 \cdot 3 = 720$

4.3. Автоматизация решения задач транспортного типа средствами EXCEL

Рассмотрим пример решения средствами EXCEL следующей транспортной задачи:

Производственное объединение имеет в своем составе четыре филиала, которые производят однородную продукцию соответственно в количествах 12, 28, 11 и 14 единиц. Эту продукцию получают три потребителя, расположенные в разных местах. Их потребности в продукции равны соответственно 20, 30, и 15 единиц. Тарифы перевозок единицы продукции от каждого из филиалов соответствующим потребителям задаются матрицей

$$C = \begin{pmatrix} 1 & 7 & 8 \\ 12 & 4 & 6 \\ 10 & 15 & 14 \\ 5 & 13 & 16 \end{pmatrix}$$

Математическая модель для определения плана перевозок минимальной стоимости в данном случае имеет следующий вид:

$$x_{11} + 7x_{12} + 8x_{13} + 12x_{21} + 4x_{22} + 6x_{23} + 10x_{31} + 15x_{32} + 14x_{33} + 5x_{41} + \\ + 13x_{42} + 16x_{43} \rightarrow \min;$$

$$x_{11} + x_{12} + x_{13} = 12;$$

$$x_{21} + x_{22} + x_{23} = 28;$$

$$x_{31} + x_{32} + x_{33} = 11;$$

$$x_{41} + x_{42} + x_{43} = 14;$$

$$x_{11} + x_{21} + x_{31} + x_{41} = 20;$$

$$x_{12} + x_{22} + x_{32} + x_{42} = 30;$$

$$x_{13} + x_{23} + x_{33} + x_{43} = 15;$$

$$x_{ij} \geq 0, \quad i = \overline{1,4}, j = \overline{1,3}.$$

Таблица EXCEL с введенными условиями данной задачи представлена на рис. 4.1. Ввод исходных данных для транспортной задачи аналогичен вводу условий обычной задачи линейного программирования. Сначала составляется форма для ввода, затем в полученную форму вводятся исходные данные задачи (запасы, потребности, тарифы на перевозку), далее осуществляется ввод зависимости для целевой функции и зависимостей для левых частей всех ограничений.

	A	B	C	D	E	F
1	Пункты		Пункты	назначения		
2	отправления	B1	B2	B3		Запасы
3	A1				=СУММ(B3:D3)	12
4	A2				=СУММ(B4:D4)	28
5	A3				=СУММ(B5:D5)	11
6	A4				=СУММ(B6:D6)	14
7		=СУММ(B3:B6)	=СУММ(C3:C6)	=СУММ(D3:D6)		
8	Потребности	20	30	15		=СУММПРОИЗВ(B3:D6;B11:D14)
9						
10						
11		1	7	8		
12	Тарифы	12	4	6		
13		10	15	14		
14		5	13	16		

Рис. 4.1

В ячейки F3-F6 занесены запасы всех пунктов отправления, в ячейки B8-D8 - потребности пунктов назначения. Значения тарифов на перевозку записаны в ячейках B11-D14. Ячейки B3-D6 соответствуют переменным задачи и в них после решения будут занесены значения перевозимых грузов. Ячейки E3-E6, B7-D7 соответствуют левым частям ограничений. Зависимость для целевой функции заносится в ячейку F8.

Для решения задачи вызывается программа **Поиск решения**. В появившемся диалоговом окне назначается целевая функция (ячейка F8), определяется направление поиска (минимизация), вводятся адреса искомых переменных (ячейки B3-D6). Для ввода ограничений выбирается команда **Добавить**. В появившемся диалоговом окне **Добавление ограничений** вводятся ограничения $BS\$7=BS\8 , $SC\$7=SC\8 , $SD\$7=SD\8 , $ES\$3=FS\3 , $ES\$4=FS\4 , $ES\$5=FS\5 , $ES\$6=FS\6 .

Далее выбирается пункт **Параметры** и в появившемся диалоговом окне **Параметры поиска решения** устанавливается флажок **Линейная модель** и **Неотрицательные значения**. Результаты решения задачи представлены на рис. 4.2. Анализ оптимального решения транспортной задачи производится так же, как и для обычных

задач линейной оптимизации. Могут быть созданы отчеты по результатам, пределам и устойчивости.

	А	В	С	Д	Е	Ф
1	Пункты		Пункты	назначения		
2	отправления	В1	В2	В3		Запасы
3	А1	6	2	4	12	12
4	А2	0	28	0	28	28
5	А3	0	0	11	11	11
6	А4	14	0	0	14	14
7		20	30	15		
8	Потребности	20	30	15		388
9						
10						
11		1	7	8		
12	Тарифы	12	4	6		
13		10	15	14		
14		5	13	16		

Рис. 4.2.

Если в условиях задачи существует запрет на перевозку (например, из пункта A_i в пункт B_j перевозка грузов запрещена), то тариф c_{ij} приравнивается какому-либо большому числу, либо при задании оптимизационной модели вводится ограничение $x_{ij}=0$.

4.4. Лабораторная работа №3

РЕШЕНИЕ ТРАНСПОРТНЫХ ЗАДАЧ ЛИНЕЙНОЙ ОПТИМИЗАЦИИ СРЕДСТВАМИ EXCEL.

Цель работы: Получение практических навыков решения задач транспортного типа в среде EXCEL.

Лабораторное задание.

1. В соответствии с вариантом решить с использованием средств EXCEL все предлагаемые задачи (пункты 1,2,3). Сформировать отчеты, построить диаграммы. Проанализировать полученные результаты.

2. Решить транспортную задачу из пункта 1 вручную с использованием метода потенциалов.

3. Сравнить результаты, полученные при выполнении лабораторных заданий 1 и 2, сделать выводы.

Отчет по лабораторной работе должен содержать :

1. Постановки решаемых транспортных задач.
2. Математические оптимизационные модели решаемых транспортных задач.
3. Результаты решения задач средствами EXCEL (по каждой задаче отдельно). Открытую модель (задача 3) предварительно преобразовать к закрытой.
4. Сформированные отчеты, построенные диаграммы.
5. Решение задачи из п.1 вручную с использованием метода потенциалов.
6. Выводы.

Вариант 1.

1. Решить транспортную задачу

Пункты отправления	Пункты назначения				Запасы
	В1	В2	В3	В4	
A1	2	4	7	9	200
A2	5	1	8	12	270
A3	11	6	4	3	140
Потребности	120	80	250	160	

2. Решить задачу из п. 1 при дополнительных условиях: из пункта A1 в пункт В3 должно быть перевезено не более 20 единиц груза, из пункта A2 в пункт В2 перевозка груза запрещена, из пункта A3 в пункт В1 необходимо перевезти не менее 5 единиц груза.

3. Решить задачу из п. 1, если запас пункта отправления A1 составит 150 единиц, а потребность пункта назначения В4 200 единиц (открытая модель).

Вариант2.

1. Решить транспортную задачу

Пункты отправления	Пункты назначения				Запасы
	В1	В2	В3	В4	
A1	3	2	7	9	190
A2	8	3	5	10	250
A3	10	6	2	4	280
Потребности	220	100	250	150	

2. Решить задачу при следующих дополнительных условиях: из пункта A2 в пункт В4 должно быть перевезено не более 30 единиц груза, из пункта A1 в пункт В2 перевозка груза запрещена, а из пункта A3 в пункт В3 необходимо перевезти не менее 10 единиц груза.

3. Решить задачу из п. 1, если запас пункта отправления A1 составит 150 единиц, а потребность пункта назначения B4 200 единиц (открытая модель).

Вариант 3.

1. Решить транспортную задачу

Пункты отправления	Пункты назначения				Запасы
	B1	B2	B3	B4	
A1	8	6	8	3	250
A2	3	3	5	18	300
A3	13	5	7	2	130
Потребности	150	100	260	170	

2. Решить задачу при дополнительных условиях: из пункта A3 в пункт B1 должно быть перевезено не более 30 единиц груза, из пункта A2 в пункт B2 перевозка груза запрещена, а из пункта A1 в пункт B1 необходимо перевезти не менее 15 единиц груза.

3. Решить задачу из п. 1, если запас пункта отправления A1 составит 200 единиц, а потребность пункта назначения B4 200 единиц (открытая модель).

Вариант 4

1. Решить транспортную задачу

Пункты отправления	Пункты назначения				Запасы
	B1	B2	B3	B4	
A1	4	5	7	8	300
A2	5	6	3	11	470
A3	8	9	6	3	230
Потребности	220	180	340	260	

2. Решить задачу при следующих дополнительных условиях: Из пункта А2 в пункт В2 должно быть перевезено не более 50 единиц груза, из пункта А2 в пункт В3 перевозка груза запрещена, а из пункта А3 в пункт В4 необходимо перевезти не менее 25 единиц груза.

3. Решить задачу из п. 1, если запас пункта отправления А3 составит 150 единиц, а потребность пункта назначения В2 300 единиц (открытая модель).

Вариант 5.

1. Решить транспортную задачу

Пункты отправления	Пункты назначения				Запасы
	В1	В2	В3	В4	
А1	11	6	4	3	100
А2	5	1	8	12	270
А3	2	4	7	9	280
Потребности	140	180	200	130	

2. Решить задачу при следующих дополнительных условиях: Из пункта А1 в пункт В1 должно быть перевезено не более 30 единиц груза, из пункта А2 в пункт В4 перевозка груза запрещена, а из пункта А3 в пункт В3 необходимо перевезти не менее 35 единиц груза.

3. Решить задачу из п. 1, если запас пункта отправления А2 составит 150 единиц, а потребность пункта назначения В1 200 единиц (открытая модель).

Вариант 6.

1. Решить транспортную задачу

Пункты отправления	Пункты назначения				Запасы
	В1	В2	В3	В4	
A1	4	5	8	6	200
A2	7	1	2	2	150
A3	9	3	9	6	300
Потребности	150	170	100	230	

2. Решить задачу при дополнительных условиях: Из пункта А2 в пункт В3 должно быть перевезено не более 60 единиц груза, из пункта А1 в пункт В3 перевозка груза запрещена, а из пункта А2 в пункт В3 необходимо перевезти не менее 25 единиц груза.

3. Решить задачу из п. 1, если запас пункта отправления А3 составит 250 единиц, а потребность пункта назначения В2 200 единиц (открытая модель).

Вариант 7.

1. Решить транспортную задачу

Пункты отправления	Пункты назначения				Запасы
	В1	В2	В3	В4	
A1	9	4	4	3	120
A2	6	2	3	11	250
A3	3	5	8	2	280
Потребности	240	80	150	180	

2. Решить задачу при дополнительных условиях: Из пункта А3 в пункт В1 должно быть перевезено не более 20 единиц груза, из пункта А1 в пункт В4 перевозка груза запрещена, а из пункта А2 в пункт В3 необходимо перевезти не менее 45 единиц груза.

3. Решить задачу из п. 1, если запас пункта отправления А2 составит 150 единиц, а потребность пункта назначения В4 200 единиц (открытая модель).

Вариант 8.

1. Решить транспортную задачу

Пункты отправления	Пункты назначения				Запасы
	В1	В2	В3	В4	
А1	12	5	3	2	110
А2	7	2	9	11	280
А3	3	5	6	7	260
Потребности	120	200	230	100	

2. Решить задачу при дополнительных условиях: Из пункта А2 в пункт В3 должно быть перевезено не более 50 единиц груза, из пункта А1 в пункт В4 перевозка груза запрещена, а из пункта А3 в пункт В3 необходимо перевезти не менее 30 единиц груза.

3. Решить задачу из п. 1, если запас пункта отправления А3 составит 150 единиц, а потребность пункта назначения В1 200 единиц (открытая модель).

Вариант 9.

1. Решить транспортную задачу

Пункты отправления	Пункты назначения				Запасы
	В1	В2	В3	В4	
А1	15	7	5	4	130
А2	4	2	9	10	250
А3	3	3	6	5	270
Потребности	160	140	200	150	

2. Решить задачу при следующих дополнительных условиях: Из пункта А3 в пункт В3 должно быть перевезено не более 40 единиц груза, из пункта А1 в пункт В4 перевозка груза запрещена, а из пункта А3 в пункт В1 необходимо перевезти не менее 15 единиц груза.

3. Решить задачу из п. 1, если запас пункта отправления А2 составит 150 единиц, а потребность пункта назначения В4 200 единиц (открытая модель).

Вариант 10.

1. Решить транспортную задачу

Пункты отправления	Пункты назначения				Запасы
	В1	В2	В3	В4	
А1	12	7	6	4	200
А2	6	2	8	11	170
А3	3	5	2	8	280
Потребности	120	200	100	230	

2. Решить задачу при следующих дополнительных условиях: Из пункта А1 в пункт В2 должно быть перевезено не более 20 единиц груза, из пункта А2 в пункт В4 перевозка груза запрещена, а из пункта А3 в пункт В3 необходимо перевезти не менее 15 единиц груза.

3. Решить задачу из п. 1, если запас пункта отправления А3 составит 150 единиц, а потребность пункта назначения В2 200 единиц (открытая модель).

5. РЕШЕНИЕ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

5.1. Модели дискретной оптимизации

Задачей дискретной оптимизации называется задача оптимизации, в которой на варьируемые параметры наложено требование дискретности:

$$\begin{aligned} f(X) &\rightarrow \min(\max) \\ g_i(X) &\geq (\leq, =) b_i, \quad i = \overline{1, m} \\ x_j &\in D_j, \quad j = \overline{1, n}, \end{aligned} \quad (5.1)$$

где D_j – множество допустимых значений каждого параметра x_j . При этом предполагается, что хотя бы один параметр x_j может принимать дискретные значения.

Частными случаями задач дискретной оптимизации являются задачи *целочисленной* и *булевой* оптимизации. В задачах целочисленной оптимизации варьируемые параметры могут принимать только целочисленные значения. В задачах булевой оптимизации $x_j \in \{0, 1\}$.

Рассмотрим наиболее распространенные прикладные модели дискретной оптимизации, использующиеся при решении задач автоматизированного проектирования и управления.

Задача о ранце

Имеются предметы n видов, причем для каждого предмета j -го вида ($j=1, n$) известны его объем $a_j > 0$ и стоимость $c_j > 0$. Необходимо определить такой набор предметов, суммарный объем которых не превышал бы заданного числа b , а суммарная ценность была бы максимальной. Эта задача интерпретируется как задача загрузки ранца объема b и называется *одномерной задачей о ранце*.

Введем целочисленные переменные x_j ($j = \overline{1, n}$), значения которых характеризует количество предметов j -го вида, помещенных в ранец. Тогда математическая модель данной задачи имеет вид:

$$\begin{aligned} \sum_{j=1}^n c_j x_j &\rightarrow \max; \\ \sum_{j=1}^n a_j x_j &\leq b; \\ x_j &\geq 0, \quad j = \overline{1, n}, \quad x_j - \text{целые}. \end{aligned}$$

Если ограничениями могут быть не только объем ранца, но и другие его характеристики b_i , то получим *многомерную задачу о ранце*:

$$\begin{aligned} \sum_{j=1}^n c_j x_j &\rightarrow \max; \\ \sum_{j=1}^n a_{ij} x_j &\leq b_i, \quad i = \overline{1, m}; \\ x_j &\geq 0, \quad j = \overline{1, n}, \quad x_j - \text{целые} \end{aligned}$$

В случае, если количество предметов j -го вида ограничено и равно d_j , к задаче добавляется ограничение $x_j \leq d_j$. Если $d_j=1$, то получим *задачу о ранце с булевыми переменными*. Тогда $x_j \in \{0, 1\}$, причем $x_j = 1$, если j -й предмет помещен в ранец, $x_j = 0$ в противном случае.

К задаче о ранце сводится широкий класс задач дискретной оптимизации с ограниченными ресурсами.

Задача о назначениях

Имеется n исполнителей и n видов работ, которые они могут выполнять. Пусть c_{ij} – производительность i -го исполнителя при

выполнении j работы. Каждый исполнитель может выполнять один вид работ, а каждый вид работ может быть выполнен одним исполнителем. Требуется таким образом распределить исполнителей по видам работ, чтобы общая производительность была максимальной.

Введем альтернативные переменные x_{ij} :

$$x_{ij} = \begin{cases} 1, & \text{если } i - \text{й исполнитель назначается на } j - \text{ю работу} \\ 0 & \text{в противном случае} \end{cases}$$

Тогда математическая модель задачи имеет вид:

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \max;$$

$$\sum_{j=1}^n x_{ij} = 1, i = \overline{1, n};$$

$$\sum_{i=1}^n x_{ij} = 1, j = \overline{1, n};$$

$$x_{ij} \in \{0, 1\}.$$

Иногда задача о назначениях формулируется как задача минимизации, если в качестве c_{ij} выбирается время, затраченное i -м исполнителем на выполнение j -й работы.

Необходимо заметить, что условие целочисленности переменных в задаче о назначениях можно не накладывать, т. к. эта задача является частным случаем транспортной задачи, и при целочисленности правых частей ограничений целочисленность решения обеспечивается автоматически.

К задаче о назначениях сводится широкий круг задач дискретной оптимизации (распределение исполнителей по видам работ, закрепление за станками операций, распределение задач между различными ЭВМ, назначение претендентов на вакантные должности при формировании штатного расписания и т. д.).

Задача коммивояжера

Имеются n пунктов с заданными расстояниями d_{ij} между i -м и j -м пунктами. Необходимо составить оптимальный маршрут из условия минимизации суммарного расстояния для коммивояжера, выходящего из пункта 1, который должен побывать во всех пунктах ровно по одному разу и вернуться в исходный пункт.

Введем альтернативные бинарные переменные:

$$x_{ij} = \begin{cases} 1, & \text{если переезд из } i - \text{го города в } j - \text{й входит в маршрут} \\ 0 & \text{в противном случае} \end{cases}$$

Условия минимизации общего расстояния, а также прибытия в каждый пункт и выезда из него ровно по одному разу могут быть выражены следующим образом:

$$\sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \rightarrow \min;$$

$$\sum_{j=1}^n x_{ij} = 1, i = \overline{1, n};$$

$$\sum_{i=1}^n x_{ij} = 1, j = \overline{1, n};$$

Однако необходимо обеспечить непрерывность маршрута, чтобы набор звеньев, входящих в маршрут, образовывал единую цепочку (например, при $n=8$ цепочка $(1,2) - (2,6) - (6,4) - (4,8) - (8,5) - (5,3) - (3,7) - (7,1)$), а не состоял бы из несвязанных цепочек (например, $(1,2) - (2,6) - (6,1)$ и $(3,8) - (8,7) - (7,5) - (5,4) - (4,3)$). Для устранения замкнутых циклов (подобходов), включающих количество вершин меньше n , в модель вводятся n дополнительных переменных $u_i \geq 0$ ($i = \overline{1, n}$) и n^2 дополнительных ограничений:

$$U_i - U_j + (n - 1)x_{ij} \leq n - 2, i = \overline{2, n}, j = \overline{2, n}.$$

Действительно, пусть маршрут включает несколько цепочек. Тогда существует цепочка, начинающаяся и заканчивающаяся в начальном пункте, но включающая n_1 звеньев, где $n_1 < n$. Просуммировав эти неравенства вдоль такой цепочки (т. е. при $x_{ik}=1$), получим бессмысленное неравенство $n_1(n-1) \leq n_1(n-2)$ (все u_i и u_j при суммировании взаимно уничтожаются). Покажем теперь, что для маршрута, исключающего подходы, это неравенство выполняется, т. е., можно найти значения u_i , удовлетворяющие дополнительным ограничениям. Положим $u_i = p$, если город i посещается коммивояжером на p -м шаге, $p = \overline{2, n}$. Отсюда следует, что $u_i - u_j \leq n - 2 \quad \forall i, j = \overline{2, n}$. Таким образом, ограничения выполняются для всех $x_{ij} = 0$. При $x_{ij} = 1$ они превращаются в равенства:

$$U_i - U_j + (n - 1)x_{ij} = p - (p + 1) + n - 1 = n - 2.$$

Задача коммивояжера имеет широкий круг приложений. К ней сводятся задачи оптимальной маршрутизации (выбор маршрутов движения транспорта, минимизация расстояния, проходимого исполнительным механизмом станка с ЧПУ и др.), задачи проектирования электрических и вычислительных сетей, задачи определения последовательности обработки деталей на станках с условием минимизации времени переналадок и т. д.

Задача о покрытии

Пусть имеется n предметов, каждый из которых обладает некоторым числом признаков из заданного множества m признаков, а в совокупности эти n предметов обладают всеми m признаками. Необходимо выбрать минимальное число предметов, которые в совокупности обладали бы m признаками. Условия задачи задаются матрицей A с элементами

$$a_{ij} = \begin{cases} 1, & \text{если } j\text{-й предмет обладает } i\text{-м признаком} \\ 0 & \text{в противном случае} \end{cases}$$

Введем бинарные переменные:

$$x_j = \begin{cases} 1, & \text{если } j\text{-й предмет выбран} \\ 0 & \text{в противном случае} \end{cases}$$

Тогда математическая модель задачи имеет следующий вид:

$$\begin{aligned} \sum_{j=1}^n x_j &\rightarrow \min; \\ \sum_{j=1}^n a_{ij} x_j &\geq 1, i = \overline{1, m}; \\ x_j &\in \{0, 1\}, \quad j = \overline{1, n} \end{aligned}$$

Каждое i -е ограничение в данном случае показывает, что должен быть выбран хотя бы один предмет, обладающий i -м признаком.

Если каждому j -му предмету приписывается вес c_j , может быть сформулирована *взвешенная задача о покрытии*:

$$\begin{aligned} \sum_{j=1}^n c_j x_j &\rightarrow \min \\ \sum_{j=1}^n a_{ij} x_j &\geq 1, \quad i = \overline{1, m}, \\ x_j &\in \{0, 1\}, \quad j = \overline{1, n} \end{aligned}$$

Если требуется найти минимальное число таких предметов, что i -м признаком обладает не менее λ_i предметов из указанного набора, получаем *задачу о кратном покрытии*:

$$\begin{aligned} \sum_{j=1}^n x_j &\rightarrow \min \\ \sum_{j=1}^n a_{ij} x_j &\geq \lambda_i, \quad i = \overline{1, m}, \\ x_j &\in \{0, 1\}, \quad j = \overline{1, n} \end{aligned}$$

Задача о разбиении

Задача о разбиении аналогична задачи о покрытии с тем отличием, что признаки у различных предметов не должны совпадать:

$$\begin{aligned}\sum_{j=1}^n x_j &\rightarrow \min; \\ \sum_{j=1}^n a_{ij}x_j &= 1, i = \overline{1, m}; \\ x_j &\in \{1, 0\}, \quad j = \overline{1, n}\end{aligned}$$

Задача о взвешенном разбиении формулируется в виде:

$$\begin{aligned}\sum_{j=1}^n c_j x_j &\rightarrow \min; \\ \sum_{j=1}^n a_{ij}x_j &= 1, i = \overline{1, m}; \\ x_j &\in \{1, 0\}, \quad j = \overline{1, n}\end{aligned}$$

5.2. Методы решения задач дискретной оптимизации

5.2.1. Метод отсечений Гомори

Рассмотрим задачу целочисленной линейной оптимизации:

$$\begin{aligned}F = \sum_{j=1}^n c_j x_j &\rightarrow \min \\ \sum_{j=1}^n a_{ij}x_j &= b_i, \quad i = \overline{1, m} \\ x_j &\geq 0, \quad j = \overline{1, n} \\ x_j &- \text{целые}, \quad j = \overline{1, k}\end{aligned} \tag{5.2}$$

Если все переменные задачи должны принимать целочисленные значения ($k = n$), задача называется *полностью целочисленной*.

Если требование целочисленности предъявляется не ко всем переменным ($k < n$), задача называется *частично целочисленной*.

Решение задач целочисленной линейной оптимизации с использованием метода отсечений сводится к решению специальным образом построенных задач линейной оптимизации: $P_0, P_1, \dots, P_s$. Первая задача P_0 образуется из исходной задачи целочисленной оптимизации Z путем отбрасывания ограничений целочисленности переменных. Каждая последующая задача $P_1, \dots, P_s$ получается из предыдущей путем добавления к ее условиям дополнительного линейного ограничения-неравенства, называемого *правильным отсечением* (или сечением). При этом r -м сечением называется линейное ограничение, вводимое в задачу P_{r-1} и соответствующее следующим двум условиям:

1. Любое целочисленное решение задачи P_{r-1} ему удовлетворяет.
2. Хотя бы одно из нецелочисленных решений задачи P_{r-1} ему не удовлетворяет (отсекается).

Таким образом, к исходной задаче P_0 последовательно добавляются дополнительные ограничения (отсечения), не исключающее целочисленных допустимых точек и отсекающие нецелочисленные решения решаемой задачи целочисленной оптимизации.

Методы отсечения различаются между собой способами формирования дополнительных ограничений. Наиболее распространенным является *метод отсечения Гомори*.

Пусть задача P_{r-1} решена симплексным методом и ее решение не удовлетворяет условиям целочисленности.

Введем обозначения:

$[a]$ – целая часть числа a , т.е. наибольшее целое число, не превосходящее a ;

$\{a\} = a - [a]$ – дробная часть числа a ;

Bas – множество индексов базисных переменных из последней симплекс-таблицы;

S – номер строки из последней симплексной таблицы, которой соответствует переменная с наибольшей дробной частью из оптимального решения. При определении данного номера S в столбце правых частей ограничений выбирается наибольшее значение $\{b_S\}$.

Тогда *I сечение Гомори* для решения полностью целочисленной задачи запишется в виде:

$$\sum_{j \notin \text{Bas}} \{a_{Sj}\}x_j \geq \{b_S\}$$

Основные шаги метода Гомори:

1. Определяется оптимальное решение задачи (5.2) без учета условий целочисленности. При этом ограничения на целочисленность переменных отбрасываются, и задача решается обычным симплекс-методом.

2. Если полученное решение является целочисленным, работа алгоритма заканчивается (оптимальное решение найдено), в противном случае осуществляется переход к шагу 3.

3. На основании последней симплекс-таблицы, полученной при решении ЗЛП, формируется сечение Гомори (I или II).

4. Дополнительное ограничение, построенное на предыдущем этапе, добавляется к условиям решаемой задачи. Для этого данное ограничение преобразуется в уравнение

$$\sum_{j \notin \text{Bas}} \{a_{Sj}\}x_j - x_d = \{b_S\}$$

где x_d – дополнительная неотрицательная переменная.

Затем полученное уравнение умножается на -1

$$- \sum_{j \notin \text{Bas}} \{a_{Sj}\}x_j + x_d = -\{b_S\}$$

и добавляется к последней симплекс-таблице. Далее полученная задача решается двойственным симплекс-методом.

5. Переход к шагу 1.

Двойственный симплекс-метод отличается от обычного симплекс-метода правилами выбора ведущей строки и ведущего столбца. Сначала осуществляется выбор ведущей строки. При этом в столбце правых частей ограничений выбирается наименьший отрицательный элемент, и строка, которой он соответствует, объявляется ведущей строкой.

$$b_s = \min b_i \quad \forall i: b_i < 0$$

Затем осуществляется выбор ведущего столбца. При этом для s -й ведущей строки определяются отношения $\frac{\Delta_j}{a_{sj}}$ для $a_{sj} < 0$, среди них выбираются минимальное, и столбец, которому оно соответствует, объявляется ведущим

$$\frac{\Delta_k}{a_{sk}} = \min_{a_{sj} < 0} \frac{\Delta_j}{a_{sj}}$$

Пересчет элементов симплекс-таблицы и остальные шаги осуществляются так же, как и в обычном симплекс-методе.

Пример 5.1. Решить задачу целочисленного линейного программирования методом Гомори:

$$\begin{aligned} & -x_1 - 4x_2 \rightarrow \min \\ & \begin{cases} -x_1 + 2x_2 + x_3 = 2 \\ 3x_1 + 2x_2 + x_4 = 6 \end{cases} \\ & x_1, x_2, x_3, x_4 \geq 0 \\ & x_1, x_2, x_3, x_4 \text{ целые.} \end{aligned}$$

Решение. Решаем данную задачу симплекс-методом без учета условий целочисленности.

x _Б	c _Б	B	-1	-4	0	0
			x ₁	x ₂	x ₃	x ₄
x ₃	0	2	-1	<u>2</u>	1	0
x ₄	0	6	3	2	0	1
		0	1	4	0	0
x ₂	-4	1	-1/2	1	1/2	0
x ₄	0	4	<u>4</u>	0	-1	1
		-4	3	0	-2	0
x ₂	-4	3/2	0	1	3/8	1/8
x ₁	-1	1	1	0	-1/4	1/4
		-7	0	0	-5/4	-3/4

Полученное решение ($x_1^* = 1$; $x_2^* = 3/2$; $x_3^* = 0$; $x_4^* = 0$) не является целочисленным. Поэтому переходим к построению дополнительного ограничения. Для этого необходимо выбрать ту строку таблицы, которой соответствует переменная с наибольшей дробной частью. В данном случае единственной такой строкой является первая строка ($S = 1$). Записываем первое сечение Гомори:

$$\left\{\frac{3}{8}\right\}x_3 + \left\{\frac{1}{8}\right\}x_4 \geq \left\{\frac{3}{2}\right\} \quad \text{или} \quad \frac{3}{8}x_3 + \frac{1}{8}x_4 \geq \frac{1}{2}$$

Умножив левую и правую часть на 2, получим:

$$\frac{3}{4}x_1 + \frac{1}{4}x_4 \geq 1$$

Введем в левую часть дополнительную переменную x_5 и умножим полученное равенство на (-1) . В результате будем иметь следующее равенство:

$$-\frac{3}{4}x_3 - \frac{1}{4}x_4 + x_5 = -1$$

Добавим полученное ограничение к последней симплекс-таблице и решим полученную задачу с использованием двойственного симплекс-метода

x _Б	c _Б	B	-1	-4	0	0	0
			x ₁	x ₂	x ₃	x ₄	x ₅
x ₂	-4	3/2	0	1	3/8	1/8	0
x ₁	-1	1	1	0	-1/4	1/4	0
x ₅	0	-1	0	0	<u>-3/4</u>	-1/4	1
		-7	0	0	-5/4	-3/4	0
x ₂	-4	1	0	1	0	0	1/2
x ₁	-1	4/3	1	0	0	1/3	-1/3
x ₃	0	4/3	0	0	1	1/3	-4/3
		-16/3	0	0	0	-1/3	-5/3

В результате получено следующее оптимальное решение:

$$x_1^* = 4/3; x_2^* = 1; x_3^* = 4/3; x_4^* = 0$$

Полученное решение не является целочисленным, поэтому опять переходим к построению сечения Гомори. Так как значения дробных частей у переменных x_1 и x_3 одинаковы, строку для построения дополнительного ограничения выбираем произвольно (например, $S = 2$). Тогда ограничение записывается в виде

$$\left\{\frac{1}{3}\right\}x_4 + \left\{-\frac{1}{3}\right\}x_5 \geq \left\{\frac{4}{3}\right\} \text{ или } \frac{1}{3}x_4 + \frac{2}{3}x_5 \geq \frac{1}{3}$$

После соответствующих преобразований получим

$$-x_4 - 2x_5 + x_6 = -1$$

Добавим данное ограничение к последней симплекс-таблице и решим полученную задачу двойственным симплекс-методом

x _Б	c _Б	В	-1	-4	0	0	0	
			x ₁	x ₂	x ₃	x ₄	x ₅	x ₆
x ₂	-4	1	0	1	0	0	1/2	0
x ₁	-1	4/3	1	0	0	1/3	-1/3	0
x ₃	0	4/3	0	0	1	1/3	-4/3	0
x ₆	0	-1	0	0	0	<u>-1</u>	-2	1
		-16/3	0	0	0	-1/3	-5/3	0
x ₂	-4	1	0	1	0	0	-1/2	0
x ₁	-1	1	1	0	0	0	-1	1/3
x ₃	0	1	0	0	1	0	-6	1/3
x ₄	0	1	0	0	0	1	2	-1
		-5	0	0	0	0	-1	-1/3

Полученное целочисленное решение является оптимальным решением исходной задачи: $x_1^* = 1$; $x_2^* = 1$; $x_3^* = 1$; $x_4^* = 1$.

5.2.2. Метод ветвей и границ

Метод ветвей и границ относится к комбинаторным методам решения задач дискретной оптимизации и может использоваться для решения как *линейных*, так и *нелинейных* задач. Он основан на использовании конечности множества допустимых вариантов и замене полного перебора сокращенным направленным перебором. При

этом полного перебора удастся избежать за счет отбрасывания неперспективных множеств вариантов, т. е. тех множеств, где заведомо нет оптимального решения. В методе ветвей и границ все множество допустимых вариантов последовательно делится на все меньшие подмножества. При этом вычисляются оценки (границы), которые позволяют сделать вывод о том, какое из полученных подмножеств может быть отброшено при условии сохранения подмножества, содержащего оптимальное решение. Для иллюстрации работы метода ветвей и границ используется дерево, называемое *деревом перебора* или *деревом вариантов* (рис. 5.1). Каждой вершине дерева вариантов соответствует какая-либо подзадача Z_i исходной задачи Z .

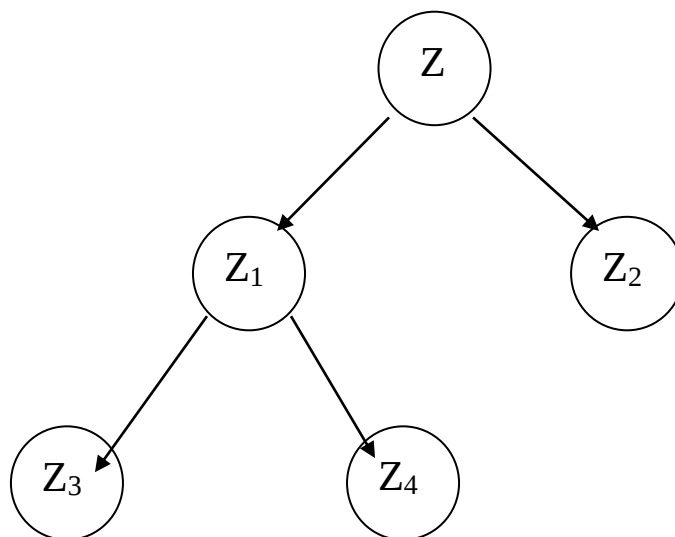


Рис. 5.1

Перед рассмотрением метода ветвей и границ введем некоторые определения.

1. Для задачи $\min_{X \in D} f(X)$ *подзадачами* будем называть задачи следующего вида: $\min_{X \in \tilde{D}} f(X)$, где $\tilde{D} \subset D$. Таким образом, подзадача – это задача с той же целевой функцией $f(X)$, но с меньшей допустимой областью, которая входит в исходную область D .

Так как при сужении допустимой области результат решения задачи ухудшается, то в общем случае выполняется следующее соотношение между оптимальными решениями исходной задачи и ее подзадач:

$$\min_{X \in D} f(X) \leq \min_{X \in \tilde{D}} f(X).$$

2. *Релаксацией* задачи (или подзадачи) будем называть переход от задачи $\min_{X \in D} f(X)$ к задаче с той же целевой функцией, но с некоторой допустимой областью D' , включающей исходную область D в качестве подмножества: $\min_{X \in D'} f(X)$, где $D \subset D'$ (одним из примеров релаксации является, в частности, отбрасывание условий целочисленности в задаче целочисленной оптимизации).

Таким образом, при релаксации задачи происходит расширение ее допустимой области. Поэтому выполняется следующее соотношение между оптимальными решениями исходной задачи и ее релаксации:

$$\min_{X \in D'} f(X) \leq \min_{X \in D} f(X).$$

Таким образом, оптимальное решение задачи $\min_{X \in D'} f(X)$ будет являться нижней границей оптимального решения исходной задачи.

3. *Ветвлением* будем называть разбиение допустимой области на непересекающиеся подмножества и получение таким образом новых оптимизационных подзадач. При этом каждой подзадаче будет соответствовать вершина дерева вариантов.

4. *Нижней оценкой* (нижней границей) μ_i оптимального значения целевой функции в вершине i будем называть оптимальное решение задачи, полученной в результате релаксации подзадачи Z_i .

5. *Верхней оценкой* (верхней границей) ν_i оптимального решения в вершине i будем называть значение целевой функции, обеспечиваемое некоторым допустимым решением подзадачи Z_i (напри-

мер, для задачи целочисленной оптимизации это целочисленное допустимое решение).

6. Значение минимальной верхней оценки в вершинах дерева вариантов называется *рекордом*.

7. Прозондированные вершины – это вершины дерева вариантов, в которых дальнейшее ветвление невозможно или нецелесообразно.

Пусть решается задача целочисленной минимизации целевой функции $\min_{X \in D_Z} f(X)$, которую в дальнейшем обозначим Z . При этом

D_Z – область допустимых решений исходной задачи, включающая условия целочисленности переменных.

Рассмотрим основные этапы метода ветвей и границ при решении данной задачи.

1. Вычисление нижней оценки оптимального решения исходной задачи Z . С этой целью осуществляется релаксация задачи Z (например, в результате отбрасывания условий целочисленности) и формируется расширенная допустимая область D . Построенная задача решается с использованием одного из методов непрерывной оптимизации (например, с использованием симплекс-метода при линейности целевой функции и ограничений), после чего осуществляется анализ полученного результата X^* .

1.1. Если оптимальное решение X^* целочисленное, то оно является оптимальным решением исходной задачи Z .

1.2. Если X^* нецелочисленное, то необходимо продолжить решение. При этом $\mu_Z = f(X^*)$ – нижняя граница оптимального целочисленного решения исходной задачи.

2. Ветвление, т. е. разбиение области допустимых решений D_Z исходной задачи Z на два подмножества: D_{Z_1} и D_{Z_2} . При этом исходная задача Z разбивается на две подзадачи: Z_1 и Z_2 .

Для организации ветвления может быть использована следующая схема. Пусть x_j^* – какая-либо нецелочисленная переменная из полученного на предыдущем этапе оптимального решения X^* . При этом область D_{z_1} формируется путем добавления к области D_z следующего дополнительного ограничения:

$$x_j \leq [x_j^*], \text{ где } [x_j^*] \text{ – целая часть числа } x_j^*.$$

Область D_{z_2} образуется путем добавления к D_z ограничения: $x_j \geq [x_j^*] + 1$. При этом формируется задача Z_1 с допустимой областью D_{z_1} и задача Z_2 с допустимой областью D_{z_2} .

3. Вычисление нижних оценок μ_{z_1} и μ_{z_2} для подзадач Z_1 и Z_2 . Для этого осуществляется релаксация подзадач Z_1 и Z_2 и их решение. При этом $\mu_{z_1} = f(X_1^*)$, $\mu_{z_2} = f(X_2^*)$, где X_1^* и X_2^* – оптимальные решения полученных задач.

Если какая-либо из подзадач не имеет решения, ее нижняя оценка считается равной бесконечности.

4. Из подзадач Z_1 и Z_2 выбирается подзадача для ветвления.

5. Ветвление выбранной подзадачи (аналогично п. 2). После ветвления снова вычисляются нижние оценки для полученных подзадач и т. д.

Процесс ветвления продолжается до получения целочисленного оптимального решения для одной из подзадач, например Z_k . При этом значение целевой функции на полученном целочисленном решении задачи Z_k представляет собой верхнюю границу оптимального решения: $v_{z_k} = f(X_k^*)$. Минимальная из полученных верхних границ будет являться рекордом.

После получения верхних границ определяются прозондированные вершины. Вершина является прозондированной, если она удовлетворяет одному из следующих условий:

- 1) оптимальное решение в этой вершине целочисленно;
- 2) значение нижней границы в этой вершине больше текущего значения рекорда (дальнейшее ветвление этой вершины нецелесообразно, так как в процессе ветвления значение целевой функции будет только увеличиваться);
- 3) подзадача, соответствующая данной вершине, не имеет допустимых решений. При этом в случаях 2) и 3) прозондированные вершины отбрасываются.

Процесс продолжается до тех пор, пока все вершины не будут прозондированы. Прозондированная вершина, соответствующая целочисленному решению с наилучшим значением целевой функции, определяет оптимальное решение исходной задачи.

Пример 5.2. Для иллюстрации основных принципов метода ветвей и границ рассмотрим следующую задачу целочисленной линейной оптимизации:

$$\begin{aligned}
 & -7x_1 - 3x_2 \rightarrow \min \\
 & 5x_1 + 2x_2 \leq 20 \\
 & 8x_1 + 4x_2 \leq 38 \\
 & x_1, x_2 \geq 0, \quad x_1, x_2 - \text{целочисленные}
 \end{aligned}$$

Начальный шаг решения этой задачи методом ветвей и границ состоит в нахождении решения задачи линейного программирования, получаемой при отбрасывании условий целочисленности x_1 и x_2 . Обозначим эту задачу через ЛП-1. Решим задачу ЛП-1 с использованием симплекс-метода. Ее оптимальное решение имеет вид: $x_1^* = 1$; $x_2^* = 7.5$, оптимальное значение целевой функции $f_1 = -29.5$. Найденное значение f_1 представляет собой *нижнюю границу* истинного оптимального целочисленного значения, поскольку при выполнении требования целочисленности x_2 значение оптимальное значений целевой функции может лишь увеличиться.

Следующий шаг метода ветвей и границ состоит в разбиении задачи ЛП-1 на две подзадачи, первая из которых (ЛП-2) образуется в результате добавления к задаче ЛП-1 ограничения $x_2 \leq 7$, а вторая (ЛП-3) – в результате добавления ограничения $x_2 \geq 8$.

ЛП-2	ЛП-3
$-7x_1 - 3x_2 \rightarrow \min$	$-7x_1 - 3x_2 \rightarrow \min$
$5x_1 + 2x_2 \leq 20$	$5x_1 + 2x_2 \leq 20$
$8x_1 + 4x_2 \leq 38$	$8x_1 + 4x_2 \leq 38$
$x_2 \leq 7$ (новое ограничение)	$x_2 \geq 8$ (новое ограничение)
$x_1, x_2 \geq 0$	$x_1, x_2 \geq 0$

Оптимальное решение задачи ЛП-3: $x_1^* = 0.75$; $x_2^* = 8$, где $f_2 = -29.25$.
 Оптимальное решение задачи ЛП-2: $x_1^* = 1.2$; $x_2^* = 7$, где $f_3 = -29.4$.

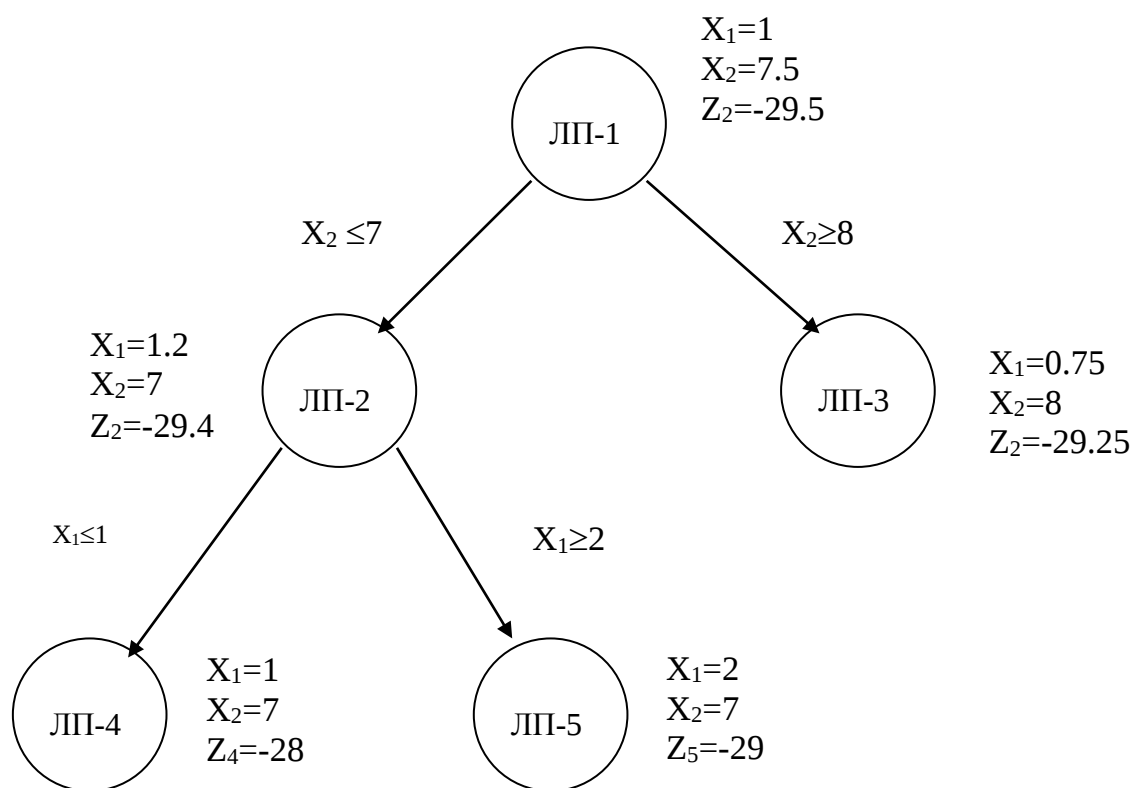


Рис. 5.2.

Для дальнейшего ветвления выберем задачу ЛП-2, т.к. значение целевой функции в ней меньше. К задаче ЛП-2 добавим ограничения $x_1 \leq 1$, (задача ЛП-4) и $x_1 \geq 2$ (задача ЛП-5). Результаты решения задачи ЛП-4: $x_1^* = 1$; $x_2^* = 7$, где $f_4 = -28$. Результаты решения задачи ЛП-5: $x_1^* = 2$; $x_2^* = 5$, где $f_5 = -29$.

Таким образом, получены два допустимых (целочисленных) решения исходной задачи целочисленного программирования, которые представляют собой *верхние границы* оптимального решения, причем значение $f_5 = -29$ будет являться *рекордом*. Даже если исходная задача имеет другие целочисленные решения, значение целевой функции в них не может быть больше -29.

Вершины ЛП-4 и ЛП-5 являются *прозондированными* (так как в них получено целочисленное решение, а в процессе дальнейшего ветвления оно может лишь ухудшаться). Ветвление вершины ЛП-3 также нецелесообразно в связи с тем, что целевая функция исходной задачи может принимать только целочисленные значения (т. к. все переменные и коэффициенты целочисленны), а при дальнейшем ветвлении ее значения будут увеличиваться (т. е., станут больше -29). Следовательно, оптимальное решение задачи:

$$x_1^* = 2; x_2^* = 5, \text{ где } f(x_1^*, x_2^*) = -29.$$

5.3. Решение задач дискретной оптимизации с использованием EXCEL

Задачи целочисленного линейного программирования в среде EXCEL решаются аналогично обычным задачам линейного программирования. Основное отличие заключается во вводе требования целочисленности.

Рассмотрим решение задачи производственного планирования, рассмотренной в лабораторной работе №2, скорректировав исход-

ные данные так, как представлено на рис. 5.3. и добавив к ней условие целочисленности всех переменных.

	ПЕРЕМЕННЫЕ						
	B1	B2	B3	B4	B5	B6	B7
имя	прод1	прод2	прод3	прод4			
значение							
нижн. гр							
верх. гр							
целочисл	целое	целое	целое	целое	ЦФ	напр	
коэф.в ЦФ	60	70	120	130	0	макс	
	ОГРАНИЧЕНИЯ						
вид				левая часть	знак	правая часть	
трудовые	1	1	1	1	0 <=	16	
сырье	6	5	4	3	0 <=	110	
финансы	4	6	10	13	0 <=	150	

Рис. 5.3

Дальнейшая работа выполняется в диалоговом окне **Поиск решения**. В нем устанавливается целевая ячейка (F7), изменяемые ячейки (B3:E3), указывается направление поиска (максимизация). Далее выбирается команда **Добавить** и в появившемся диалоговом окне **Добавление ограничения** вводятся ограничения: $F9 \leq H9$, $F10 \leq H10$, $F11 \leq H11$. Определяются требования целочисленности переменных (для этого в данном диалоговом окне для всех переменных из списка условных операторов (между ссылкой и ограничением) выбирается *целое*). Условия неотрицательности переменных можно ввести в диалоговом окне **Параметры поиска решения**. В окне **Параметры поиска решения** устанавливается также флажок **Линейная модель**, после чего осуществляется запуск задачи на решение. Результаты представлены на рис. 5.4.

Если решить ту же задачу без условий целочисленности переменных, получим следующий результат:

Прод1=1.67, Прод2=0, Прод3=14.33, Прод4=0. Получаемая прибыль при этом будет равна 1820. Это показывает, что требование целочисленности, как и любое другое дополнительное требование, ухудшает целевую функцию.

	A	B	C	D	E	F	G	H
1	Переменные							
2	имя	прод1	прод2	прод3	прод4			
3	значение	1	1	14	0			
4	нижн. гр.							
5	верхн. гр.					ЦФ	напр	
6	коэф. в ЦФ	60	70	120	130	1810	макс	
7	Ограничения							
8	вид					левая часть	знак	правая часть
9	трудовые	1	1	1	1	16	<=	16
10	сырье	6	5	4	3	67	<=	110
11	финансы	4	6	10	13	150	<=	150

Рис. 5.4.

Исходя из требований целочисленности, для целочисленных задач возможен вызов только одного **Отчета по результатам**. Отчет по результатам для рассматриваемой задачи представлен на рис. 29. Отчет состоит из трех таблиц.

Таблица 1 содержит сведения о целевой функции. В столбце "Исходно" приведены значения целевой функции до начала вычислений.

Таблица 2 содержит сведения об изменяемых ячейках E13:E16. Здесь приведены значения искомым переменных, полученные в результате оптимального решения задачи.

Таблица 3 показывает результаты оптимального решения для ограничений и граничных условий.

В таблице для ограничений:

"формула" - зависимости для ограничений;

"значение" - величина использованного ресурса;

"разница" - величина неиспользованного ресурса.

Если ресурс используется полностью, то в графе "Состояние" указывается связанное. При неполном использовании ресурса в этой графе указывается не связан.

Величины для граничных условий состоят из двух групп. В первой группе - величины, аналогичные значениям для ограничений, при этом **Разница** показывает разность между найденным зна-

чением переменной и заданным граничным условием; во второй группе - значения для требований целочисленности, поэтому все они являются связанными.

Целевая ячейка (Макс)					
Ячейка	Имя	Исходно	Результат		
\$F\$7	коэф. в ЦФ ЦР	0	1810		
Изменяемые ячейки					
Ячейка	Имя	Исходно	Результат		
\$B\$3	значение прод1	0	1		
\$C\$3	значение прод2	0	1		
\$D\$3	значение прод3	0	14		
\$E\$3	значение прод4	0	0		
Ограничения					
Ячейка	Имя	Значение	Формула	Состояние	Разница
\$F\$10	трудовые левая часть	16	\$F\$10<=\$H\$10	связанное	0

Рис. 5.5.

В диалоговом окне **Параметры поиска решения** есть возможность вводить различные значения **Допустимого отклонения**. Увеличение величины допустимого отклонения ухудшает точность решения, но ускоряет его. По умолчанию в Excel установлено допустимое отклонение, равное 5 %, что вполне приемлемо для большинства практических задач.

5.4. Лабораторная работа №4

РЕШЕНИЕ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ СРЕДСТВАМИ EXCEL

Цель работы: Получение практических навыков формирования математических моделей прикладных задач дискретной оптимизации и их решения средствами EXCEL.

Лабораторные задания.

1. Выбрать задачу из задания 1 в соответствии с вариантом, составить ее математическую модель и решить с использованием средств EXCEL. Сформировать отчеты по устойчивости, результатам и пределам, построить диаграммы.

2. В соответствии с условием задачи провести параметрический анализ и определить, как влияет на результаты изменение условий задачи.

3. В соответствии с вариантом выбрать и решить в EXCEL задачу из задания 2.

Отчет по лабораторной работе должен содержать :

1. Постановки задач в соответствии с вариантом.
2. Математические оптимизационные модели.
3. Краткое описание основных этапов решения в среде EXCEL.
4. Результаты решения задач (итоговые таблицы в EXCEL).
5. Отчеты по результатам.
6. Построенные диаграммы.
7. Результаты параметрического анализа с соответствующими выводами (для задания 1).
8. Выводы.

Задание 1

Для организации вычислительной лаборатории могут быть использованы четыре типа ЭВМ. Объем оперативной памяти первой ЭВМ t_1 Гбайт, второй - t_2 Гбайт, третьей - t_3 Гбайт, четвертой - t_4 Гбайт. Стоимость первой ЭВМ a_1 тыс.руб, второй - a_2 тыс. руб, третьей - a_3 тыс. руб, четвертой - a_4 тыс. руб. Потребляемая мощность первой ЭВМ b_1 Вт, второй - b_2 Вт, третьей - b_3 Вт, четвертой - b_4 Вт. Первая ЭВМ занимает площадь c_1 кв.м., вторая c_2 кв.м., третья c_3 кв.м., четвертая c_4 кв.м. Определить, сколько ЭВМ и какого типа надо выбрать, чтобы обеспечить максимальный объем оперативной памяти при ограниченных стоимостных ресурсах d_1 тыс. руб, энергетических ресурсах d_2 Вт и ограниченной площади d_3 кв.м.

Вариант	1,2	3,4	5,6	7,8	9,10
t_1	4	8	16	8	8
t_2	16	4	32	4	4
t_3	8	16	4	32	16
t_4	4	4	8	16	8
a_1	11	8	25	12	14
a_2	20	9	28	15	11
a_3	13	21	10	28	18
a_4	15	12	15	16	15
b_1	300	220	220	220	220
b_2	350	220	300	220	350
b_3	400	300	350	300	400
b_4	450	350	400	350	220
c_1	1.5	1	1.3	1.5	1.5
c_2	1.5	1.3	2.2	2	1.5
c_3	2	1.7	2	2	2
c_4	2.5	1.6	1.5	2.5	2
d_1	140	130	150	110	120
d_2	2500	2000	2200	2000	2400
d_3	15	20	22	23	25

Определить, каким образом изменится решение, если:

1. Стоимостные ресурсы уменьшатся до 100 тыс. руб.
2. Площадь лаборатории увеличится до 30 кв.м.
3. Энергетические ресурсы уменьшатся до 1500 Вт.

КАЖДЫЙ ПУНКТ РЕШИТЬ ОТДЕЛЬНО. СЦЕНАРИИ ДЕЛАТЬ НЕ НАДО.

Задание 2

Задача 1 (Вариант 1,10). На пяти станках различных типов можно выполнять пять операций по обработке детали. При этом за каждым из станков может быть закреплена лишь одна операция, а одна и та же операция может выполняться только одним станком. Время выполнения каждой из операций на каждом из станков задается матрицей

$$C = \begin{pmatrix} 2 & 7 & 1 & 5 & 3 \\ 4 & 3 & 2 & 7 & 8 \\ 6 & 2 & 8 & 4 & 2 \\ 8 & 4 & 7 & 3 & 9 \\ 2 & 2 & 1 & 4 & 5 \end{pmatrix}$$

Необходимо составить такое распределение выполняемых операций между станками, при котором суммарные затраты времени на обработку детали являются минимальными.

Указание. Данная задача формализуется в виде задачи о назначениях.

Задача 2. (Вариант 2,9) Пусть некоторое количество информации хранится в 6 массивах (файлах) длины $c_j, j = \overline{1,6}$, причем на каждую единицу информации отводится по крайней мере один массив. Получена заявка на 4 типа единиц информации.

Задана матрица A с элементами

$$a_{ij} = \begin{cases} 1, & \text{если в } j\text{-м массиве находится } i\text{-я информация} \\ 0 & \text{в противном случае} \end{cases}$$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Необходимо определить подмножество массивов минимальной длины, необходимых для удовлетворения заявки.

Величины $c_j, j = \overline{1,6}$ выбрать самостоятельно

Указание. Данная задача формализуется в виде задачи о покрытии. При этом вводятся альтернативные переменные:

$$x_j = \begin{cases} 1, & \text{если } j\text{-й массив выбирается} \\ 0 & \text{в противном случае} \end{cases}$$

Задача 3 (вариант 3,8). При передаче сообщений по каналу с шумом вероятности соответствия принимаемой j -й буквы передаваемой i -й букве задаются матрицей:

$$P = \begin{pmatrix} 0,1 & 0,2 & 0,5 & 0,7 & 0,1 \\ 0,4 & 0,3 & 0,1 & 0,8 & 0,7 \\ 0,6 & 0,2 & 0,6 & 0,9 & 0,2 \\ 0,4 & 0,7 & 0,3 & 0,3 & 0,8 \\ 0,3 & 0,2 & 0,5 & 0,4 & 0,2 \end{pmatrix}$$

Необходимо каждой букве передаваемого алфавита поставить в соответствие букву принимаемого таким образом, чтобы сумма вероятностей правильности принимаемых букв была бы максимальна.

Указание. Данная задача формализуется в виде задачи о назначениях. При этом вводятся альтернативные переменные:

$$x_{ij} = \begin{cases} 1, & \text{если } j\text{-я принимаемая буква соответствует } i\text{-й передаваемой} \\ 0 & \text{в противном случае} \end{cases}$$

Задача 4 (вариант 4,7) Для проверки 4 неисправностей в системе может быть использовано 6 типов диагностических тестов, каждый из которых характеризуется временем работы $c_j, j = \overline{1,6}$.

$$c_1 = 1, c_2 = 5, c_3 = 2, c_4 = 3, c_5 = 1, c_6 = 6$$

Задана матрица T с элементами

$$t_{ij} = \begin{cases} 1, & \text{если } j\text{-й Тест может использоваться для проверки } i\text{-й неисправности} \\ 0 & \text{в противном случае} \end{cases}$$

$$T = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Необходимо выбрать тесты, обеспечивающие проверку всех неисправностей за минимальное время.

Указание. Данная задача формализуется в виде задачи о покрытии. При этом вводятся альтернативные переменные:

$$x_j = \begin{cases} 1, & \text{если } j\text{-й Тест выбирается} \\ 0 & \text{в противном случае} \end{cases}$$

Задача 5 (Варианты 5,6). Имеется 5 исполнителей и 5 видов работ, которые они могут выполнять. Задана матрица C , каждый элемент которой c_{ij} – время, которое затрачивает i -й исполнитель на выполнение j работы. Каждый исполнитель может выполнять один вид работ, а каждый вид работ может быть выполнен одним исполнителем. Требуется таким образом распределить исполнителей по видам работ, чтобы общее время выполнения работ было минимальным.

$$C = \begin{pmatrix} 2 & 5 & 6 & 8 & 3 \\ 1 & 2 & 1 & 6 & 5 \\ 7 & 12 & 4 & 5 & 8 \\ 9 & 3 & 2 & 2 & 1 \\ 3 & 7 & 1 & 4 & 5 \end{pmatrix}$$

Указание. Данная задача формализуется в виде задачи о назначениях.

6. НЕЛИНЕЙНАЯ ОПТИМИЗАЦИЯ

6.1. Постановка задач нелинейной оптимизации

В общем виде задача нелинейной оптимизации может быть сформулирована следующим образом:

$$\begin{aligned} &\text{extr } f(X); \\ &g_i(X) \geq (\leq, =) b_i, \quad i = \overline{1, m}; \\ &x_j \geq 0, \quad j = \overline{1, n}, \end{aligned}$$

где хотя бы одна из функций $f(x)$, $g_i(x)$ является нелинейной. Здесь $X = (x_1, \dots, x_n)$ – вектор варьируемых параметров. На переменные x_j также могут накладываться ограничения: $d_j \leq x_j \leq D_j$, где d_j и D_j – нижняя и верхняя границы для каждой j -й переменной. Задача без ограничений называется задачей *безусловной* оптимизации, задача с ограничениями – задачей *условной* оптимизации.

6.2. Методы решения задач нелинейной оптимизации

6.2.1. Преобразование задач с ограничениями к задачам безусловной оптимизации

Одним из наиболее распространенных и часто используемых на практике приемов решения задач нелинейной оптимизации с ограничениями является их преобразование к задачам безусловной оптимизации. При этом для учета прямых и функциональных ограничений могут быть использованы различные подходы.

Рассмотрим задачу нелинейной оптимизации с прямыми и функциональными ограничениями:

$$\begin{aligned}
f(X) &\rightarrow \min \\
g_i(X) &\geq (\leq, =) b_i, \quad i = \overline{1, m} \\
x_j^{\min} &\leq x_j \leq x_j^{\max}, \quad j = \overline{1, n}
\end{aligned} \tag{6.1}$$

Прямые ограничения на варьируемые параметры можно исключить из данной задачи с помощью замены переменных:

$$x_j = V_j(z_j), \quad j = \overline{1, n}, \quad -\infty \leq z_j \leq \infty,$$

где $Z = (z_1, \dots, z_n)$ – вектор новых варьируемых параметров.

Тогда целевая функция задачи может быть переформулирована следующим образом:

$$f(X) = f(V_1(Z), \dots, V_n(Z)) = \hat{f}(Z) \rightarrow \min.$$

Аналогичным образом переформулируются и функциональные ограничения:

$$g_i(X) = g_i(V_1(Z), \dots, V_n(Z)) = \hat{g}_i(Z) \geq (\leq, =) b_i$$

Переформулированная в новых переменных задача примет вид:

$$\begin{aligned}
\hat{f}(Z) &\rightarrow \min \\
\hat{g}_i(Z) &\geq (\leq, =) b_i, \quad i = \overline{1, m}
\end{aligned}$$

При этом прямые ограничения из задачи исключаются, так как новые переменные $Z = (z_1, \dots, z_n)$ могут изменяться в любых пределах. В случае, если границы изменения всех варьируемых параметров одинаковы, замена переменных может иметь вид:

$$x_j = V(z_j), \quad j = \overline{1, n}, \quad -\infty \leq z_j \leq \infty$$

Рассмотрим теперь основные подходы к учету *функциональных ограничений*. Пусть решается задача минимизации целевой функции $f(X)$ с функциональными ограничениями, представленными в виде системы неравенств (6.2) и равенств (6.3):

$$f(X) \rightarrow \min$$

$$g_i(X) \geq b_i, \quad i = \overline{1, m} \quad (6.2)$$

$$h_j(X) = d_j, \quad j = \overline{1, r} \quad (6.3)$$

Для учета функциональных ограничений обычно используется *метод штрафных функций*. При этом осуществляется переход от исходной целевой функции $f(X)$ к следующей функции:

$$P(X, \gamma) = f(X) + \gamma \cdot S(X).$$

Здесь $S(X)$ – *штрафная функция (функция штрафа)*, отличная от нуля вне допустимой области D ; $\gamma > 0$ – коэффициент штрафа, значение которого может быть постоянным или меняться на различных итерациях. Во втором случае параметр $\gamma_k > 0$ настраивается в ходе оптимизационного процесса (k – номер итерации).

В результате решение задач с ограничениями сводится к решению последовательности задач безусловной оптимизации вспомогательной функции $P(X)$. При этом штрафная функция $S(X)$ формируется таким образом, чтобы при нарушении ограничений задачи производился некоторый “штраф” за их нарушение. При решении задачи минимизации “штраф” заключается в том, что к целевой функции прибавляется некоторое положительное число, “отбрасывая” тем самым оптимизационный процесс от оптимальной точки.

Существует два метода построения штрафных функций:

- метод *внутренних* штрафных функций (барьерных функций);
- метод *внешних* штрафных функций.

Метод *внутренних штрафных функций* предназначен для учета *только ограничений-неравенств* и характеризуется следующей основной функцией штрафа:

$$S(X) = \sum_{i=1}^m \frac{1}{g_i(X) - b_i}.$$

При этом предполагается, что ограничения-равенства (4.3) в задаче отсутствуют. Тогда целевая функция оптимизационной задачи примет вид:

$$P(X, \gamma) = f(X) + \gamma \cdot \sum_{i=1}^m \frac{1}{g_i(X) - b_i} \quad \text{или} \quad P(X, \gamma) = f(X) + \gamma_k \cdot \sum_{i=1}^m \frac{1}{g_i(X) - b_i}.$$

При этом параметр $\gamma_k > 0$ выбирается таким образом, чтобы его значения стремились к нулю при $k \rightarrow \infty$ (k – номер итерации).

При использовании внутренних штрафных функций поиск оптимума следует начинать с внутренней точки допустимой области, то есть с точки, в которой все ограничения выполнены как строгие неравенства. При выходе на границу допустимой области значение штрафной функции $S(X)$ (штраф) будет бесконечным, следовательно, оптимизационный процесс никогда не выйдет за пределы допустимой области. Недостатком данного метода является то, что он не позволяет решать задачи с ограничениями-равенствами. Кроме того, для их использования необходимо знать начальную допустимую точку. В этой связи более целесообразным является использование внешних штрафных функций.

С помощью метода *внешних штрафных функций* учитываются как ограничения-неравенства $g_i(X) \geq b_i$, так и ограничения-равенства $h_j(X) = d_j$. Наибольшее распространение получили штрафные функции следующего вида:

$$S(X) = \sum_{i=1}^m (\min\{0; g_i(X) - b_i\})^2 + \sum_{j=1}^r (h_j(X) - d_j)^2, \quad (6.4)$$

$$S(X) = \sum_{i=1}^m |\min\{0; g_i(X) - b_i\}| + \sum_{j=1}^r |h_j(X) - d_j|. \quad (6.5)$$

При этом если ограничения не нарушаются, т. е. $g_i(X) \geq b_i$ и $h_j(X) = d_j$, то $S(X)=0$. Если ограничения нарушаются, то величина “штрафа” зависит от степени нарушения ограничения.

Может быть использована также комбинированная штрафная функция, в которой ограничения-неравенства учитываются с помощью внешних штрафных функций, а ограничения-равенства – с помощью внутренних (барьерных) функций.

$$S(X) = \sum_{i=1}^m \frac{1}{g_i(x) - b_i} + \sum_{j=1}^r (h_j(X) - d_j)^2.$$

Таким образом, задачи с ограничениями с помощью замены переменных и методов штрафных функций могут быть преобразованы к задачам безусловной оптимизации. После соответствующих преобразований полученные задачи решаются с использованием методов безусловной оптимизации.

6.2.2. Принципы построения и классификация методов безусловной оптимизации

Рассмотрим задачу безусловной оптимизации:

$$f(X) = f(x_1, \dots, x_n) \rightarrow \min_{X \in R^n}$$

Алгоритмы безусловной оптимизации составляют основу алгоритмического обеспечения, предназначенного для решения оптимизационных задач. Они представляют собой итерационные процедуры, реализующие последовательное приближение к искомому экстремуму:

$$X^{k+1} = X^k + \alpha_k H^k,$$

где k – номер итерации, H^k – направление движения на k -й итерации, α_k – величина шага в данном направлении. При этом

$$X^{k+1} = \begin{pmatrix} x_1^{k+1} \\ \dots \\ x_n^{k+1} \end{pmatrix}, \quad X^k = \begin{pmatrix} x_1^k \\ \dots \\ x_n^k \end{pmatrix}, \quad H^k = \begin{pmatrix} h_1^k \\ \dots \\ h_n^k \end{pmatrix}.$$

Процесс поиска начинается с некоторой начальной точки X^0 , которая называется *начальным приближением* и задается пользователем. Получаемая в процессе поиска последовательность точек $X^0, X^1, \dots, X^k$ называется *траекторией поиска* и должна сходиться к оптимальной точке X^* .

В зависимости от способа выбора направления поиска методы безусловной оптимизации делятся на методы 0, 1 и 2-го порядка.

Методы нулевого порядка – методы, в которых для определения направления поиска используются только значения целевой функции. Производные в данном случае не вычисляются. Этот класс алгоритмов называется еще *методами прямого поиска* или *поисковыми методами оптимизации*. К ним относятся метод переменного многогранника и различные алгоритмы покоординатной оптимизации и случайного поиска. Особенностью методов поисковой оптимизации является то, что они могут быть использованы при алгоритмическом задании критериев оптимальности и ограничений.

Методы первого порядка – методы, в которых для определения направления поиска используются первые производные целевой функции по управляемым параметрам. Эти методы называют также *градиентными*. В качестве направления поиска на каждой итерации в них выбирается градиент (в задачах максимизации) или антиградиент (в задачах минимизации) оптимизируемой функции.

Методы второго порядка – методы, в которых для определения направления используются вторые производные целевой функции. К этому классу относят метод Ньютона и его модификации.

6.2.3. Методы нулевого порядка

Методы оптимизации нулевого порядка (поисковые методы оптимизации) используют в процессе оптимизации *только значения*

целевой функции. Все они являются эвристическими и реализуют различные стратегии направленного перебора.

В зависимости от способа выбора направления поиска методы нулевого порядка делятся на следующие основные группы:

1. *Методы покоординатной оптимизации*. В данных методах поиск осуществляется вдоль координатных направлений.

2. *Алгоритмы переменного многогранника*. В эту группу входит метод переменного многогранника Нелдера-Мида и его модификации. Методы базируются на стратегии поиска, заключающейся в построении в n -мерном пространстве многогранника из $n+1$ -й вершины и перемещении его в направлении оптимальной точки с помощью трех основных операций: отражения, растяжения и сжатия.

3. *Методы случайного поиска*. При использовании данных методов направление поиска на каждой итерации выбирается случайным образом. Методы отличаются друг от друга способами генерации случайных векторов, а также стратегиями анализа текущей информации и перестройки шага поиска.

Рассмотрим основные методы покоординатной оптимизации. Остальные классы методов подробно рассмотрены в [5].

В методах покоординатной оптимизации в качестве направлений поиска выбираются координатные оси пространства варьируемых параметров. При этом на каждой k -й итерации выполняется n шагов циклического покоординатного спуска из текущей точки X^k вдоль каждой из n координатных осей. Покоординатный спуск сводится к поочередному изменению переменных вдоль одной из осей:

$$X^{k+1} = X^k + \lambda_i^k \cdot e_i, \quad i = 1 \dots n,$$

где e_i – i -й координатный n -мерный вектор, у которого i -й элемент равен 1, а остальные элементы равны 0:

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix} \quad e_2 = \begin{pmatrix} 0 \\ 1 \\ \dots \\ 0 \end{pmatrix} \quad \dots \quad e_n = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 1 \end{pmatrix},$$

λ_i^k – длина шага поиска по i -й переменной на k -ой итерации.

Таким образом, в координатной форме шаги циклического спуска имеют следующий вид:

$$\begin{pmatrix} x_1^{k+1} \\ \dots \\ x_n^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^k \\ \dots \\ x_n^k \end{pmatrix} + \lambda_1^k \begin{pmatrix} 1 \\ \dots \\ 0 \end{pmatrix} \quad (i=1 - \text{первая координатная ось}),$$

...

$$\begin{pmatrix} x_1^{k+1} \\ \dots \\ x_n^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^k \\ \dots \\ x_n^k \end{pmatrix} + \lambda_n^k \begin{pmatrix} 0 \\ \dots \\ 1 \end{pmatrix} \quad (i=n - n\text{-я координатная ось}).$$

Геометрической интерпретацией траектории покоординатного поиска является ломаная, состоящая из отрезков, параллельных осям координат (рис. 6.1). На рис. 6.1 изображены также *линии уровня целевой функции* двух переменных. Каждая линия уровня соответствует некоторому постоянному значению целевой функции $f(X) = \text{const}$ (в случае большего числа переменных речь идет о *поверхностях уровня* целевой функции). Таким образом, процесс минимизации заключается в последовательном переходе от одной линии уровня $f(X) = C_i$ к линии уровня с меньшим значением целевой функции $f(X) = C_{i+1}$, ($C_{i+1} < C_i$).

Необходимо заметить, что для нелинейной целевой функции линии уровня часто имеют овражный характер. При этом целевая функция сильно меняется по одним направлениям и слабо – по дру-

гим. Такие овражные ситуации создают значительные трудности для оптимизации.

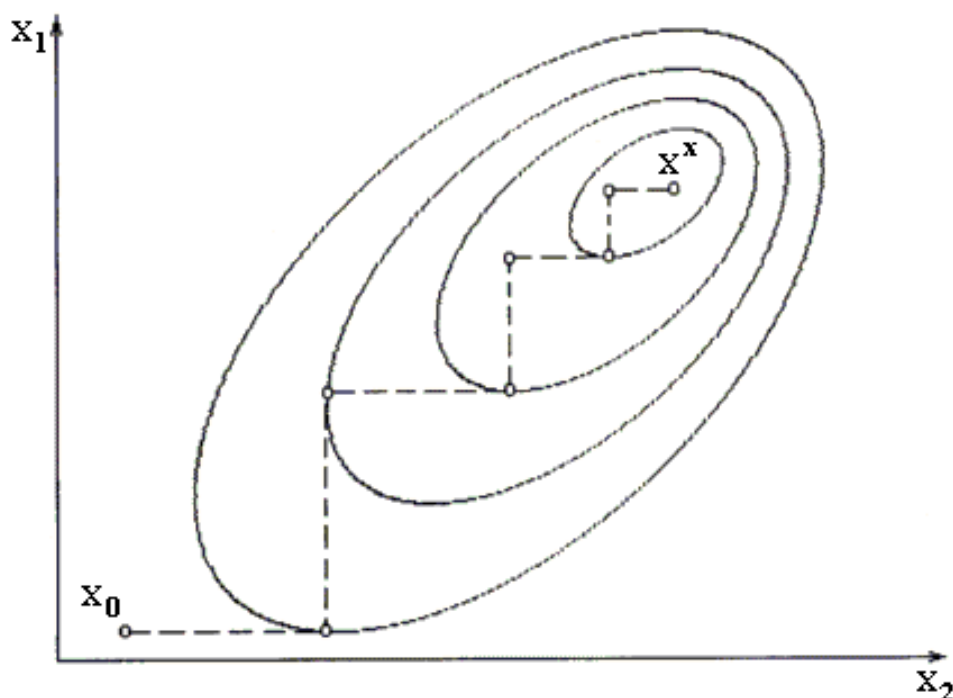


Рис. 6.1

К простейшим алгоритмам покоординатной оптимизации можно отнести *алгоритм покоординатного спуска с постоянным шагом* и *релаксационный метод Гаусса-Зейделя*. Различия между этими двумя алгоритмами заключается в способе выбора шага поиска λ_i^k .

Покоординатный спуск с постоянным шагом

В этом методе на каждой k -й итерации циклический покоординатный спуск заключается в следующем. Сначала осуществляется движение из точки X^k с шагом λ_1^k вдоль 1-й координатной оси:

$$X^k + \lambda_1^k e_1.$$

Если это приводит к уменьшению значения целевой функции: $f(X^k + \lambda_1^k e_1) < f(X^k)$, осуществляется переход в точку $X^k + \lambda_1^k e_1$. В

противном случае производится пробный шаг в противоположном направлении:

$$X^k - \lambda_1^k e_1.$$

В случае уменьшения значения целевой функции осуществляется переход в точку $X^k - \lambda_1^k e_1$, в противном случае переход в новую точку не производится. Затем рассматривается следующая координатная ось. Аналогичные действия повторяются для всех координатных осей. Если в результате движения вдоль всех координатных осей значение целевой функции не уменьшилось, осуществляется дробление шага, и циклический покоординатный спуск повторяется из точки X^k с уменьшенной длиной шага. Итерационный процесс заканчивается, когда длины шагов вдоль координатных направлений уменьшатся до некоторого числа ε .

Релаксационный алгоритм Гаусса-Зейделя

Отличие данного алгоритма от предыдущего заключается в том, что в нем на каждой k -й итерации определяются оптимальные длины шагов λ_i^k , $i = \overline{1, n}$ в результате решения вспомогательных задач одномерной оптимизации. При этом процедура поиска точки минимума X^* сводится к следующей последовательности действий.

1. Задается начальное приближение $X^0 = (x_1^0, \dots, x_n^0)$ и точность вычислений $\varepsilon > 0$. При этом $k=0$, $X^k = X^0$.

2. Осуществляется циклический покоординатный спуск из точки X^k по итерационной схеме $X^{k+1} = X^k + \lambda_i^k \cdot e_i$, $i = 1 \dots n$ с выбором оптимальной длины поискового шага λ_i^k вдоль каждого направления $i = 1 \dots n$. Эта процедура образует внутренний цикл, в процессе которого осуществляется одномерная минимизация функции $f(x_1, \dots, x_n)$ по каждой переменной x_i , $i = 1 \dots n$:

$$f(x_1^{k+1}, x_2^k, \dots, x_n^k) = \min_{x_1} f(x_1, x_2^k, \dots, x_n^k),$$

$$f(x_1^{k+1}, x_2^{k+1}, \dots, x_n^k) = \min_{x_2} f(x_1^{k+1}, x_2, \dots, x_n^k),$$

....

$$f(x_1^{k+1}, x_2^{k+1}, \dots, x_n^{k+1}) = \min_{x_n} f(x_1^{k+1}, x_2^{k+1}, \dots, x_n).$$

В результате каждого i -го шага внутреннего цикла определяется очередная координата x_i^{k+1} точки X^{k+1} . После окончания внутреннего цикла в качестве нового приближения принимается точка X^{k+1} .

3. Проверяется выполнение критерия окончания итерационного процесса. Если $|x_i^{k+1} - x_i^k| \leq \varepsilon$ для всех $i = 1 \dots n$, работа алгоритма заканчивается. В противном случае осуществляется переход к шагу 2 (при этом $k=k+1$).

Дальнейшим развитием методов покоординатной оптимизации является метод конфигураций Хука-Дживса. В данном методе этап циклического покоординатного поиска вдоль координатных осей чередуется с этапом экстраполяции, т.е. движения в направлении, соединяющем две перспективные точки X^k и X^{k+1} на двух последовательных шагах (рис. 6.2). Первый этап называется в методе Хука-Дживса *исследующим поиском вокруг базисной точки*, а второй этап – *поиском по образцу*.

На этапе исследующего поиска из базисной точки X^k осуществляется движение вдоль координатных осей в соответствии с общей схемой покоординатного спуска, в результате чего будет по-

лучена точка X^{k+1} . Если этот этап не приводит к уменьшению значения целевой функции, длина шага уменьшается и исследующий поиск повторяется.

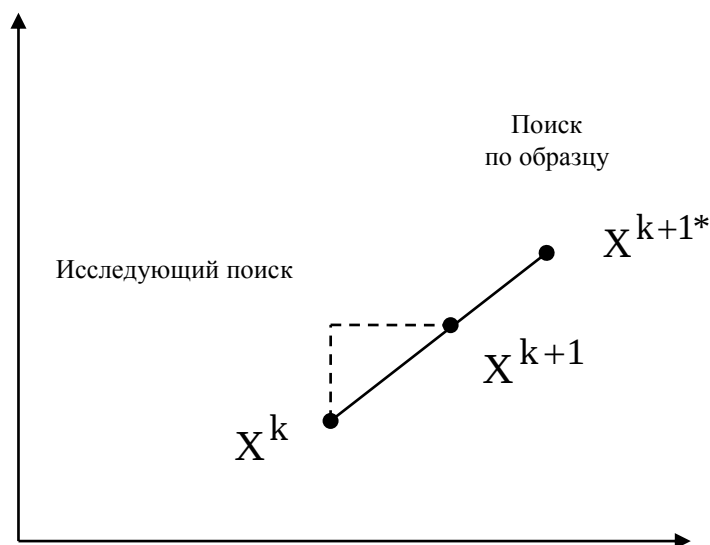


Рис. 6.2.

Если этап исследующего поиска оказывается успешным, т.е. $f(X^{k+1}) < f(X^k)$, реализуется этап поиска по образцу. При этом производится движение из точки X^{k+1} в направлении $X^{k+1} - X^k$, после чего осуществляется исследующий поиск вокруг полученной точки X^{k+1*} . Если полученное при этом значение целевой функции меньше, чем значение в точке X^{k+1} , поиск по образцу считается удачным. В противном случае (если поиск по образцу неудачен) осуществляется возврат в точку X^{k+1} и исследующий поиск продолжается вокруг нее. Алгоритм заканчивает работу, когда текущая длина шага становится меньше заданной точности ε .

6.2.4. Методы первого порядка

В методах первого порядка используются первые производные целевой функции $f(X)$. В качестве направления поиска на каждой

итерации выбирается градиент целевой функции (в задачах максимизации) или антиградиент (в задачах минимизации) Поэтому эти методы называют также *градиентными*. Итерационный процесс в градиентных методах реализуется по следующей схеме:

$$X^{k+1} = X^k + \alpha_k \nabla f(X^k) \quad (\text{максимизация}); \quad (6.6)$$

$$X^{k+1} = X^k - \alpha_k \nabla f(X^k) \quad (\text{минимизация}). \quad (6.7)$$

Здесь $\nabla f(X)$ – градиент (вектор частных производных) целевой функции:

$$\nabla f(X) = \begin{pmatrix} \frac{\partial f(X)}{\partial x_1} \\ \dots \\ \frac{\partial f(X)}{\partial x_n} \end{pmatrix}.$$

Таким образом, координатной форме градиентная процедура имеет вид (для задачи минимизации):

$$x_i^{k+1} = x_i^k - \alpha_k \frac{\partial f(X^k)}{\partial x_i}, \quad i = 1, \dots, n$$

Градиентные методы различаются в зависимости от способов выбора и корректировки величины шага α_k . Наиболее распространены два метода. Первый называется *методом с дроблением шага* и связан с проверкой на каждой итерации неравенства, обеспечивающего убывание целевой функции: $f(X^{k+1}) < f(X^k)$. В случае невыполнения неравенства осуществляется дробление шага. Вторым методом называется *методом наискорейшего спуска* и обеспечивает выбор на каждой итерации оптимальной длины шага в результате решения вспомогательной задачи одномерной оптимизации. Таким образом, этап градиентного метода (переход в новую точку поиска) чередуется с этапом определения оптимальной длины шага.

6.2.5. Методы второго порядка

Методы второго порядка используют в процессе поиска вторые производные целевой функции. К этому классу относят метод Ньютона и его модификации. Основная стратегия данных методов заключается в следующем:

$$X^{k+1} = X^k - \alpha_k [\nabla^2 f(X^k)]^{-1} \nabla f(X^k), \quad (6.9)$$

где $\nabla f(X^k)$ – градиент целевой функции $f(X)$ в точке X^k , $[\nabla^2 f(X^k)] = G(X^k)$ – матрица вторых производных (матрица Гессе) целевой функции $f(X)$ в точке X^k .

Методы данного класса отличаются друг от друга способами выбора и настройки шага α_k .

В методе Ньютона величина шага на каждой итерации выбирается равной единице: $\alpha_k = 1$. Итерационная схема поиска в данном случае имеет вид:

$$\begin{aligned} X^{k+1} &= X^k - [\nabla^2 f(X^k)]^{-1} \nabla f(X^k) \\ \text{или} \quad X^{k+1} &= X^k - G^{-1}(X^k) \nabla f(X^k). \end{aligned}$$

Метод Ньютона-Рафсона отличается от метода Ньютона тем, что в нем шаг поиска не выбирается равным единице на всех итерациях, а настраивается в ходе оптимизационного процесса. Итерационная схема поиска имеет вид:

$$X^{k+1} = X^k - \alpha_k G^{-1}(X^k) \nabla f(X^k).$$

На практике обычно используются два способа выбора длины шага. В первом случае на каждой итерации осуществляется проверка неравенства $f(X^{k+1}) < f(X^k)$ и в случае его невыполнения длина шага уменьшается. Шаг дробится до тех пор, пока неравенство не выполнится. Второй способ определения длины шага, как и в методе наискорейшего спуска, состоит в определении на каждой итерации

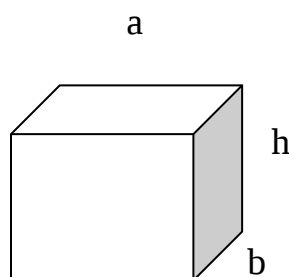
оптимальной длины шага в результате решения задачи одномерной минимизации:

$$\min_{\alpha} (X^k - \alpha G^{-1}(X^k) \nabla f(X^k)) .$$

При этом этап перехода в новую точку поиска чередуется с этапом определения оптимальной длины шага.

6.3. Решение задач нелинейной оптимизации в EXCEL

Решение задачи нелинейного программирования рассмотрим на следующем примере. Пусть требуется определить размеры бака, имеющего форму параллелепипеда заданного объема:



Объем бака :

$$V=abh.$$

Полная поверхность:

$$S=2(ab)+2(a+b)h=2(ab+(a+b)h).$$

Принимаем, что стоимость материала $C=ks$, где k -стоимость единицы площади материала.

В результате получим:

$$C=2k(ab+(a+b)h).$$

После введения рассмотренных величин сформулируем задачу оптимизации следующим образом:

$$V = abh \rightarrow \max;$$

$$2k(ab + (a + b)h) \leq C_{\text{зад}};$$

$$a, b, h \geq 0.$$

В этой постановке требуется определить размеры бака a, b, h , стоимость которого не должна превышать $C_{\text{зад}}$, чтобы его объем V был максимальным. Для решения задачи принимаем следующие значения:

$$k = 20 \text{ т. р/м}^2,$$

$$C_{\text{зад}} = 100 \text{ т.р.}$$

Тогда математическая модель примет вид:

$$V = abh \rightarrow \max;$$

$$20(ab + (a + b)h) \leq 100;$$

$$a, b, h \geq 0.$$

Решим данную задачу с использованием средств EXCEL.

Решение задачи нелинейного программирования имеет следующие отличия от решения задачи линейного программирования:

- назначаются начальные значения искомых переменных x_j^0
- в окне *Параметры поиска решения* не надо вводить *Линейная модель*.

Начальные значения x_j^0 желательно назначать близкими к ожидаемым оптимальным значениям, что ускорит решение задачи. Обязательным является требование к целевой функции которая в начальной точке должна быть не равна нулю (иначе возможно деление на ноль при вычислении ΔF_k).

Необходимо сделать формулу для ввода условий задачи (рис. 6.3), в которую далее вводятся:

- зависимости для объема и стоимости (ячейки C8, C9);
- начальные значения x_j^0 (ячейки B3, C3, D3). В данном случае в качестве начальных значений выбираются единичные;
- значение правой части ограничения (ячейка E9).

В ячейках, в которых будет представлен результат (B3:D3), перед решением задачи надо назначить число знаков после запятой. В нашем примере назначаем в ячейках 2 знака после запятой.

		ПЕРЕМЕННЫЕ		
	a	b	h	
значения	1	1	1	
нижн. гр				
		ЗАВИСИМОСТИ		
	обозначение	величина	знак	правая часть
объем	V	$B3 \cdot C3 \cdot D3$	макс	
стоимость	C	$20 \cdot (B3 \cdot C3 + (B3 + C3) \cdot D3)$	$\leq$	100

Рис. 6.3.

Далее вызывается программа *Поиск решения* и в появившемся диалоговом окне вводится ячейка для целевой функции (C8), направление поиска (максимизация), изменяемые ячейки (B3:D3). Затем выбирается пункт *Добавить* и в появившемся окне *Добавление ограничений* вводятся ограничения $B3 \geq B4$; $C3 \geq C4$; $D3 \geq D4$, $C9 \leq E9$. После ввода всех ограничений осуществляется переход в диалоговое окно *Параметры поиска решения*.

После ввода всех исходных данных и параметров производится решение задачи. Результаты представлены на рис. 6.4. После завершения поиска оптимального решения можно сформировать отчеты трех типов: *результаты*, *устойчивость* и *пределы*. Отчеты анализа по результатам и пределам аналогичны таким же отчетам для задач линейной оптимизации.

Отчет по устойчивости представлен на рис. 6.5 и состоит из двух таблиц.

В первой таблице *Изменяемые ячейки* приводятся значения для переменных:

результат решения задачи;

- нормированный градиент - величина, приводимая при выборе некоторых методов в диалоговом окне *Параметры поиска решения*;

Во второй таблице *Ограничения* приводятся значения для ограничений:

- величина стоимости;
- множитель Лагранжа, показывающий, как изменится целевая функция при изменении правой части в ограничении на единицу.

	А	В	С	Д	Е
1	Переменные				
2		а	б	г	
3	значения	1.29	1.29	1.29	
4	нижн. гр.				
5	Зависимости				
6					
7		обозначение	величина	знак	правая часть
8	объем	V	2.15	макс	
9	стоимость	С	100.00	<=	100.00

Рис. 6.4.

Изменяемые ячейки			
Ячейка	Имя	Результат значение	Нормир. градиент
\$B\$4	значения а	1.29	0.00
\$C\$4	значения б	1.29	0.00
\$D\$4	значения г	1.29	0.00
Ограничения			
		Результ.	Лагранжа
Ячейка	Имя	значение	Множитель
\$C\$10	С величина	100.00	0.03

Рис. 6.5.

Для задач нелинейной оптимизации можно произвести также параметрический анализ. Алгоритм выполнения параметрических расчетов аналогичен схеме, рассмотренной при решении задач линейной оптимизации.

6.4. Лабораторная работа №5

РЕШЕНИЕ ЗАДАЧ НЕЛИНЕЙНОЙ ОПТИМИЗАЦИИ СРЕДСТВАМИ EXCEL.

Цель работы: Получение практических навыков решения задач нелинейной оптимизации средствами EXCEL.

Лабораторные задания

1. Выбрать задание в соответствии с вариантом. Решить задачу средствами EXCEL. Сформировать отчеты, проанализировать полученные результаты.

2. Решить задачу вручную с использованием графического метода, сравнить с результатами, полученными в п.1.

3. Провести параметрический анализ, определив, как изменится решение при изменении правых частей ограничений. Сформировать итоговый сценарий.

$$\begin{array}{l} \text{Вариант 1 :} \\ x_1^2 + (x_2 - 2)^2 \rightarrow \min \\ \left\{ \begin{array}{l} 2x_1 + x_2 \geq 7 \\ x_1 + 2x_2 \geq 5 \\ x_1 \geq 1, x_2 \geq 0 \end{array} \right. \end{array}$$

$$\begin{array}{l} \text{Вариант 2:} \\ (x_1 - 5)^2 + (x_2 - 2)^2 \rightarrow \min \\ \left\{ \begin{array}{l} x_1 - 3x_2 \leq 0 \\ 2x_1 + x_2 \leq 4 \\ x_1 \geq 0, 0 \leq x_2 \leq 6 \end{array} \right. \end{array}$$

$$\begin{aligned} & (x_1 + 2)^2 + (x_2 - 2)^2 \rightarrow \min \\ \text{Вариант 3: } & \begin{cases} 7x_1 + 4x_2 \leq 43 \\ 5x_1 - x_2 \geq -4 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{aligned}$$

$$\begin{aligned} & (x_1 - 8)^2 + (x_2 - 5)^2 \rightarrow \min \\ \text{Вариант 4: } & \begin{cases} x_1 - 5x_2 \leq 0 \\ x_1 + 2x_2 \geq 16 \\ 2x_1 + x_2 \leq 11 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{aligned}$$

$$\begin{aligned} & (x_1 - 5)^2 + (x_2 - 10)^2 \rightarrow \min \\ \text{Вариант 5: } & \begin{cases} x_1 + x_2 \leq 11 \\ 4x_1 - x_2 \leq 4 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{aligned}$$

$$\begin{aligned} & (x_1 - 5)^2 + (x_2 - 8)^2 \rightarrow \min \\ \text{Вариант 6: } & \begin{cases} 2x_1 + x_2 \geq 16 \\ 5x_1 - x_2 \geq 0 \\ x_1 + 2x_2 \leq 11 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{aligned}$$

$$\begin{aligned} & (x_1 - 4)^2 + (x_2 - 10)^2 \rightarrow \min \\ \text{Вариант 7: } & \begin{cases} 3x_1 - x_2 \geq 0 \\ x_1 + x_2 \leq 8 \\ x_1 - 2x_2 \leq 0 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \end{aligned}$$

$$\text{Вариант 8: } \begin{cases} (x_1 - 7)^2 + (x_2 - 8)^2 \rightarrow \min \\ x_1 - 2x_2 \leq 0 \\ 3x_1 - x_2 \geq 0 \\ x_1 + x_2 \leq 11 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

$$\text{Вариант 9: } \begin{cases} (x_1 - 5)^2 + (x_2 - 1)^2 \rightarrow \min \\ 2x_1 - x_2 \geq -4 \\ 2x_1 - 3x_2 \geq -6 \\ x_1 + x_2 \leq 11 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

$$\text{Вариант 10: } \begin{cases} (x_1 + 3)^2 + (x_2 - 8)^2 \rightarrow \min \\ x_1 - 5x_2 \leq 0 \\ x_1 + 2x_2 \leq 11 \\ 3x_1 + x_2 \leq 18 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

Отчет по лабораторной работе должен содержать :

1. Оптимизационную модель в соответствии с вариантом.
2. Краткое описание основных этапов решения в среде EXCEL.
3. Результат решения задачи (итоговую таблицу в EXCEL).
5. Отчеты по результатам, устойчивости, пределам.
6. Решение задачи вручную графическим методом.
7. Результаты параметрического анализа.
8. Выводы.

7. РЕШЕНИЕ ЗАДАЧ ОПТИМИЗАЦИИ СРЕДСТВАМИ MATHCAD

7.1. Элементы пользовательского интерфейса MATHCAD

Mathcad – популярная система компьютерной математики, предназначенная для решения математических, в том числе оптимизационных, задач в различных предметных областях. Сегодня различные версии Mathcad являются математически ориентированными универсальными системами. Среди других систем компьютерной математики система Mathcad выделяется своим наглядным пользовательским интерфейсом.

Пользовательский интерфейс системы аналогичен интерфейсу обычных Windows-приложений (рис. 7.1), однако содержит дополнительный пункт меню **Математика**, предназначенный для решения математических задач.

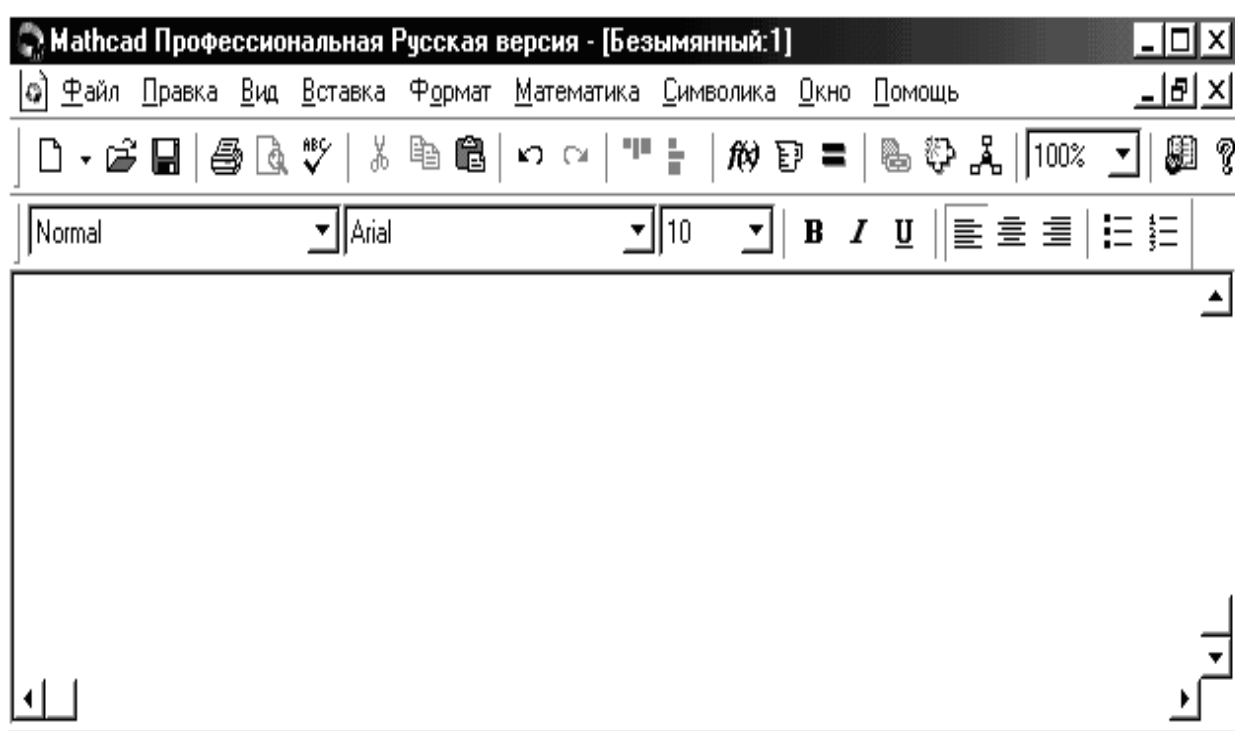
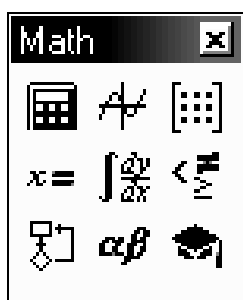


Рис. 7.1.

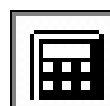
Пункт **Математика** открывает доступ к специализированным панелям математических инструментов: **Калькулятор**, **График**, **Матрица**, **Вычисление**, **Калькуляция**, **Булево**, **Программирование**, **Грек**, **Символьный**.



Каждую из специализированных панелей можно активизировать выбором соответствующего пункта меню или выбором соответствующей кнопки на обобщенной панели **Математика**.

Рассмотрим подробнее каждую специализированную панель.

Калькулятор



Панель **Калькулятор** предназначена для ввода математических знаков и имеет следующий вид:

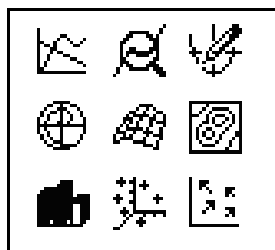
sin	cos	tan	ln	log
n!	i	x	$\sqrt{}$	$\sqrt[n]{}$
e^x	$\frac{1}{x}$	()	x^2	x^y
π	7	8	9	/
$\frac{\square}{\square}$	4	5	6	$\times$
$\div$	1	2	3	+
$:=$	.	0	-	=

Пикто- грамма	Функция	Пример	Клавиши
$n!$	Факториал целого числа n	$5!=120$	[!]
i	Мнимая единица		[1],[i]
$ x $	Модуль (абсолютное значение) числа	$x:=\cos(\pi)$ $ x =1$	[]
$\ln$ $\log$	Натуральный и десятичный логарифмы	$\ln(10)$ $\log(100)$	
e^x	Экспоненциальная функция ($e=2.718$)	e^2	[e],[^]
x^y	Степень	x^2	[^]
$\sqrt[n]{}$	Корень n -ой степени	$\sqrt[3]{8}$	[‘]
()	Скобки	$(x+2y)$	[’]
π	3.1459		[Ctrl+p]
$\sqrt{}$	Квадратный корень	$\sqrt{2}$	[N]
x^2	Квадрат числа	x^2	[^],[2]
Tan,sin,cos	Тригонометрические функции		
0..9.	Цифры и десятичная точка		[0]..[9],[.]
/	Деление	$\frac{x}{y}$	[/]
+ -	Сложение и вычитание	$x+y$ $x-y$	[+],[-]
:=	Присваивание значения	$x:=\cos(\pi)$	[:]
=	Вывод значения переменной	$x=$	[=]

График



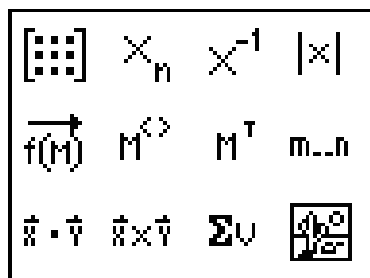
Панель **График** используется для построения графиков.



Пиктограм- ма	Функция	Клавиши
	Создать график в декартовых координатах	[@]
	Изменить масштаб графика в декартовых координатах	
	Определить координаты точки	
	Создать график в полярных координатах	[Ctrl+7]
	Создать график поверхности	[Ctrl+2]
	Создать карту линий уровня	[Ctrl+5]
	Создать трехмерную столбчатую гистограмму	
	Создать трехмерную точечную диаграмму	
	Создать векторное поле	

Матрица

Панель **Матрицы** содержит набор действий с матрицами.



Пиктограмма	Функция	Пример	Клавиши
	Ввод матрицы с указанием числа столбцов и строк	$A := \begin{pmatrix} 1 & 2 \\ 5 & 8 \end{pmatrix}$	[Ctrl+M]
	Вычисление определителя	$ A =$	[[]]
	Покомпонентное умножение матриц		[Ctrl+-]
	Сумма компонент вектора	ΣV	[Ctrl+4]
	Скалярное произведение векторов	$V \cdot P$	[*]
	Векторное произведение векторов	$V \times P$	[Ctrl+8]
	Транспонирование матрицы	$A^T =$	[Ctrl+1]
	Выбор отдельного столбца матрицы (нумерация начинается с 0)	$A^{<2>}$	[Ctrl+6]

Пиктограмма	Функция	Пример	Клавиши
	Создание матрицы, каждый элемент которой содержит значение пикселя растрового изображения		[Ctrl+T]
	Обращение матрицы	A^{-1}	[^],[-1]
	Нижний индекс	x_n	[x],[[],[n]
	Определение диапазона изменения величины	$x := m..n$	[m],[;],[n]

Греческие символы

Эта панель содержит основные символы греческого алфавита.

α	β	γ	δ	ε	ζ
η	θ	ι	κ	λ	μ
ν	ξ	$\omicron$	π	ρ	σ
τ	υ	ϕ	χ	ψ	ω
Λ	$\Bbbk$	Γ	Δ	Ξ	Σ
$\mathbb{H}$	Θ	$\mathbb{I}$	$\mathbb{K}$	$\mathbb{L}$	$\mathbb{M}$
$\mathbb{N}$	Ξ	$\mathbb{O}$	$\mathbb{P}$	$\mathbb{R}$	Σ
$\mathbb{T}$	Υ	Φ	$\mathbb{X}$	Ψ	Ω

Вычисления

Панель вычислений имеет следующий вид:

$=$	$:=$	$\equiv$
$\rightarrow$	$\mapsto$	$f x$

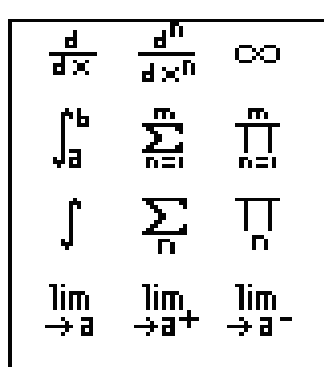
Пиктограмма	Функция	Пример	Клавиши
-------------	---------	--------	---------

	Знак равенства при численных вычислениях	$5+5=10$	[=]
	Глобальное присваивание	$y \equiv 0$	[~]
	Локальное присваивание	$z := 5$	[:]
	Знак равенства при символьных вычислениях	$\frac{dx^2}{dx} \rightarrow 2x$	[Ctrl+.]

Панель математического анализа



Панель содержит пиктограммы для выполнения основных операций математического анализа.



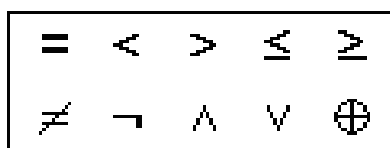
Пикто- грамма	Функция	Пример	Клавиши
	Первая производная	$\frac{dx^2}{dx} \rightarrow 2x$	[Shift+ /]
	n-я производная	$\frac{dx^2}{dx^2} \rightarrow x$	[Ctrl+Shift+ /]
	Бесконечность		[Ctrl+Shift+Z]
	Определенный интеграл	$\int_0^1 x^2 = \frac{1}{3}$	[&]

	Неопределенный интеграл	$\int_0^1 x^2 \rightarrow \frac{x^3}{3}$	[Ctrl+i]
	Суммирование по дискретному аргументу	$\sum_i x_i$	[\$]
	Суммирование	$\sum_{i=1}^{10} x_i$	[Ctrl+Shift+4]
	Произведение по дискретному аргументу	$\prod_i x_i$	[#]
	Произведение	$\prod_{i=1}^{10} x_i$	[Ctrl+Shift+3]
	Предел	$\lim_{x \rightarrow 1} \ln(x) \rightarrow 0$	[Ctrl+L]
	Правый предел	$\lim_{x \rightarrow 1^+} \ln(x) \rightarrow 0$	[Ctrl+A]
	Левый предел	$\lim_{x \rightarrow 1^-} \ln(x) \rightarrow 0$	[Ctrl+B]

Панель булевых операций



Данная панель является панелью равенств, отношений и логических операций.



Пиктограмма	Функция	Пример	Клавиши
	Равно	$x = 5$	[Ctrl++]
	Не равно	$x \neq 5$	[Ctrl+3]
	Логическое отрицание	$\neg 1 = 0$	[Ctrl+Shift+1]
	Логическое 'и'	$1 \wedge 0 = 0$	[Ctrl+Shift+7]
	Логическое 'или'	$1 \vee 0 = 1$	[Ctrl+Shift+6]
	Меньше	$x < 5$	[<]
	Больше	$x > 10$	[>]
	Меньше или рано	$x \leq 5$	[Ctrl+9]
	Больше или равно	$x \geq 10$	[Ctrl+0]
	Исключающее 'или'	$1 \oplus 0 = 1$	[Ctrl+Shift+5]

Программирование

Данная специализированная панель содержит следующие основные операторы встроенного в Mathcad языка программирования:

Add line – добавить строку в программу;

 – локальное присваивание;

if – условный оператор;

otherwise – оператор альтернативного выбора;

for – оператор цикла;

while – оператор цикла;

break – оператор прерывания;

continue – оператор продолжения;

return – оператор возврата;

on error – переход при возникновении ошибки.

Add Line	←
if	otherwise
for	while
break	continue
return	on error

Панель символьных операций



Панель символьных операций содержит основные команды символьного процессора Mathcad. С использованием этой панели можно производить различные операции с выражениями, заданными в символьной форме (преобразование, упрощение, разложение на множители, символьное решение уравнений и систем и т.д.).

→	■ →	Modifiers
float	complex	assume
solve	simplify	substitute
factor	expand	coeffs
collect	series	parfrac
fourier	laplace	ztrans
invfourier	invlaplace	invztrans
$m^T \rightarrow$	$m^{-1} \rightarrow$	$ m \rightarrow$

Пикто- грамма	Функция	Пример
→	Символьное вычисление	$\frac{d}{dt} \sin(t) \rightarrow \cos(t)$
■ →	Символьное вычисление с ключевым словом	$x^2 + 2x + 1 \text{ factor} \rightarrow (x + 1)^2$
Modifiers	Дополнительные моди- фикаторы например real, string	

Пикто- грамма	Функция	Пример
float	Численное вычисление	$\ln(2) \cdot x^2 \text{ float}, 3 \rightarrow .693 \cdot x^2$
complex	Комплексное вычисление (a и b предполагаются действительными)	$\frac{1}{a + i \cdot b} \text{ complex} \rightarrow$
assume	Символьное вычисление выражения при некотором предположении	$\sqrt{M^2} \left \begin{array}{l} \text{simplify} \\ \text{assume}, M > 0 \end{array} \right. \rightarrow M$
substitute	Замена переменной выражением	$x^2 \text{ substitute}, x = 2 \rightarrow 4$
factor	Разложение выражения на множители	$x^3 + x \text{ factor} \rightarrow x \cdot (x^2 + 1)$
expand	Раскрытие скобок и приведение подобных	$\frac{x^2 + 1}{x} \text{ expand} \rightarrow \frac{(x^2 + 1)}{x}$
coeffs	Определение коэффициентов полинома	$a \cdot x + b \text{ coeffs}, x \rightarrow \begin{pmatrix} b \\ a \end{pmatrix}$
collect	Группировка слагаемых по степеням переменной	$(3 \cdot x + y)^2 \text{ collect}, y \rightarrow y^2 + 6 \cdot x \cdot y + 9 \cdot x^2$
series	Разложение в ряд Тейлора или Лорана	$\cos(x) \text{ series}, x \rightarrow 1 - \frac{1}{2} \cdot x^2 + \frac{1}{24} \cdot x^4$
parfrac	Разложение на элементарные дроби	$\frac{2}{x^2 - 1} \text{ convert, parfrac}, x \rightarrow \frac{1}{(x - 1)} - \frac{1}{(x + 1)}$
fourier	Преобразование Фурье	$\Phi(t) \text{ fourier}, t \rightarrow \pi \cdot \text{Dirac}(\omega) - \frac{1i}{\omega}$
laplace	Преобразование Лапласа	$\Phi(t) \text{ laplace}, t \rightarrow \frac{1}{s}$

Пикто- грамма	Функция	Пример
invlaplace	Обратное преобразование Лапласа	$\frac{\omega}{s^2 + \omega^2} \text{ invlaplace, } s \rightarrow \sin(\omega \cdot t)$
ztrans	Z-преобразование	$\Phi(n) \text{ ztrans, } n \rightarrow \frac{z}{(z-1)}$
invztrans	Обратное Z-преобразование	$\frac{z}{z-1} \text{ invztrans, } z \rightarrow 1$
$M^T \rightarrow$	Транспонирование матрицы	$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad A^T \rightarrow \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$
$M^{-1} \rightarrow$	Нахождение обратной матрицы	$A^{-1} \rightarrow \begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$
$ M \rightarrow$	Нахождение определителя матрицы	$ A \rightarrow -2$
complex	Комплексное вычисление (a и b предполагаются действительными)	$\frac{1}{a + i \cdot b} \text{ complex} \rightarrow$

7.2. Основы работы в системе MATHCAD

Общение с пользователем системы Mathcad осуществляется на математически ориентированном входном языке, который является типичным языком визуального программирования. При этом используется традиционный для математической литературы способ записи функций и выражений. Математические выражения на экране компьютера представлены в общепринятой математической нотации – имеют такой же вид, как в книге, в тетради, на доске.

Алфавит входного языка Mathcad представляет совокупность разнообразных символов и слов, которые используются при задании команд. Он содержит:

- малые и большие латинские буквы;
- малые и большие греческие буквы;

- арабские цифры от 0 до 9;
- системные переменные;
- специальные знаки – операторы;
- имена встроенных функций;
- специальные знаки;
- малые и большие буквы кириллицы (при работе с русифицированными документами).

Константы

Mathcad поддерживает следующие типы констант:

- целочисленные константы (1, 5, -45 и т.д);
- действительные числа (например, 25.45 или $12.3 \cdot 10^{-5}$);
- восьмеричные числа (идентифицируются латинской буквой O);
- шестнадцатеричные числа, образуемые цифрами 0,1,2,...,A,B,C,D,E,F (имеют в конце отличительный признак в виде буквы h или H; если число начинается с буквы, перед ней надо ввести ноль);
- комплексные числа $Z = \text{Re}Z + i \cdot \text{Im}Z$ или $Z = \text{Re}Z + j \cdot \text{Im}Z$, где $\text{Re}Z$ – действительная часть комплексного числа Z , $\text{Im}Z$ – его мнимая часть, а символы i или j обозначают мнимую единицу, то есть корень квадратный из -1 ;
- системные константы, хранящие параметры системы;
- строковые константы – цепочки символов, заключенные в кавычки, например “hello”, ”2+3” (арифметические выражения в строковых константах рассматриваются как текст и не вычисляются);
- единицы измерения физических величин.

Диапазон возможных значений десятичных констант лежит в пределах от 10^{307} до 10^{-307} (это машинная бесконечность и машинный ноль).

Переменные

Переменные являются имеющими уникальные имена объектами, которые вначале не определены, а после определения могут принимать любые значения в пределах своего типа. В системе Mathcad тип переменной определяется ее значением – переменные могут быть числовыми, строковыми, символьными и т.д., поэтому тип переменной предварительно не задается.

Имена переменных (идентификаторы) могут иметь практически любую длину, и в них могут входить любые латинские и греческие буквы и цифры, однако начинаться идентификатор должен с буквы, например, x_1 , α , x_{coord} . Значения строковых переменных задаются в кавычках, например, $N := \text{''My_name''}$. При выборе имени переменной необходимо руководствоваться следующими правилами:

- в идентификаторах переменных недопустимо использование пробелов;
- идентификатор начинается с буквы;
- нельзя использовать в идентификаторах буквы русского языка;
- Малые и большие буквы в идентификаторах различаются (то есть x и X – разные переменные);
- идентификаторы должны быть уникальными, то есть они не могут совпадать с именами встроенных или определенных пользователем функций.
- некоторые спецсимволы (например, знак подчеркивания $_$) могут входить в состав идентификаторов, другие (например, знаки арифметических действий) недопустимы.

В системе Mathcad введены несколько системных переменных, имеющих зарезервированные значения. В частности, это число π и основание натурального логарифма e .

Операторы

Операторы – элементы языка, предназначенные для создания математических выражений (формул) совместно с данными, которые именуются *операндами*. К операндам относятся знаки арифметических операций (+, -,), вычисления сумм и произведений (Σ , Π , членов ряда, производных, интегралов и т.д.). После указания операндов (аргументов соответствующих операторов) операнды становятся исполняемыми программными блоками. Операторы в Mathcad вводятся с помощью шаблонов, которые в свою очередь имеют места ввода для операндов. Необходимый шаблон можно выбрать на панелях математических инструментов.

Функции

Функции Mathcad можно разделить на два класса:

1. Встроенные функции (их также называют стандартными). Они разбиты на ряд категорий, таких, например, как функции Бесселя, векторные и матричные функции, функции статистики и т.д. Список всех встроенных функций можно получить, щелкнув на кнопке **f(x)** панели инструментов. Если на имени функции установлен курсор ввода, то нажатие клавиши F1 открывает страницу справки по указанной функции. Чтобы использовать функцию в выражении, надо определить значения входных параметров в скобках после имени функции.

Наиболее часто при вводе формул возникает необходимость в элементарных функциях, таких, как синус, косинус, логарифм и т.д. Аргумент и значение элементарных функций могут быть действительными или комплексными числами. Все углы измеряются в радианах.

Примеры встроенных функций : sin, cos, root, rkfixed и т.д. (всего встроенных функций в системе MathCad около трехсот).

2. Функции, задаваемые пользователем. Они имеют следующий вид:

Имя_функции(Список параметров):=Выражение

Имя функции задается идентификатором как обычное имя переменной. В скобках после имени функции указывается список параметров, разделяемых запятыми. Выражение – это любое выражение, содержащее доступные системе операторы и функции с операндами и аргументами, указанными в списке параметров.

Документ системы MathCad называется *рабочим листом*. Он содержит следующие объекты (блоки):

- формулы;
- графические области;
- текстовые блоки.

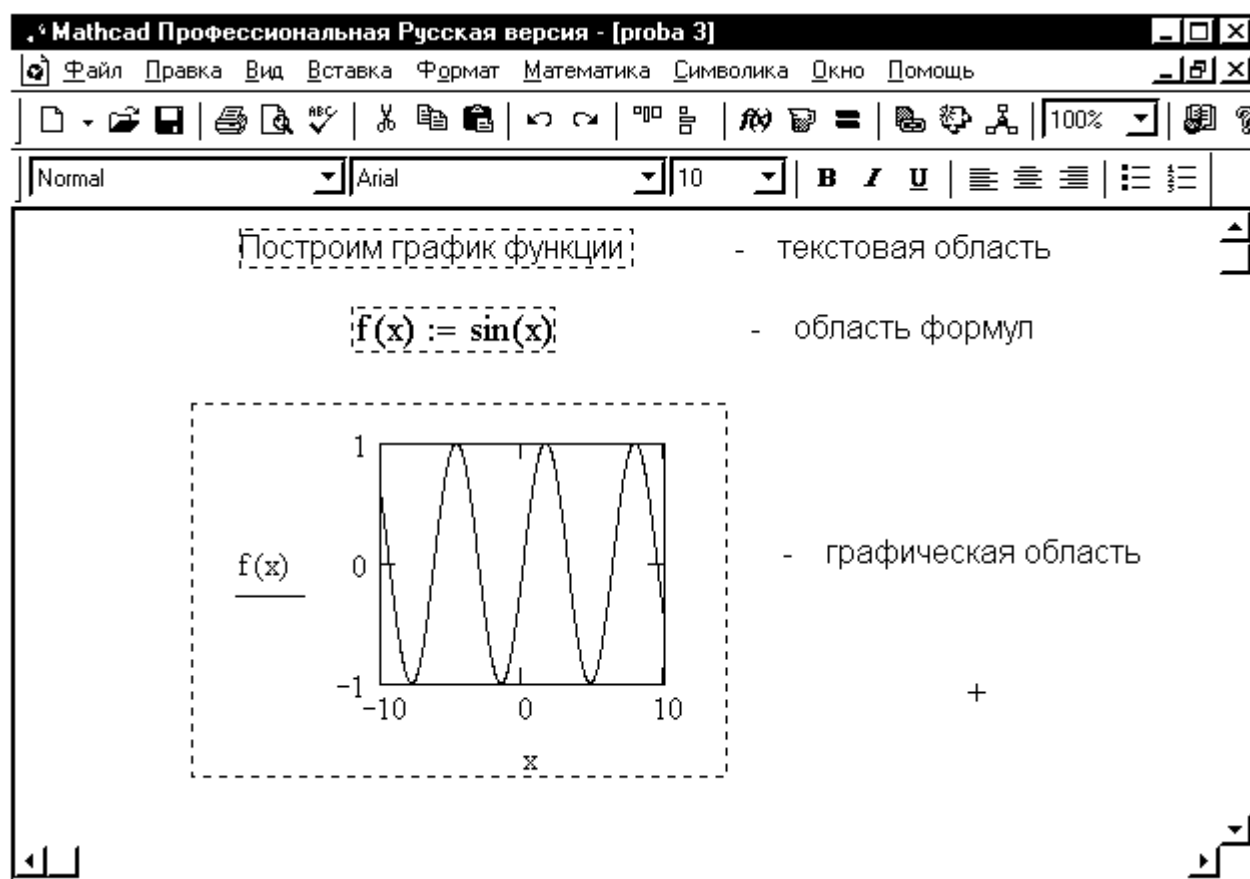
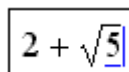


Рис. 7.2.

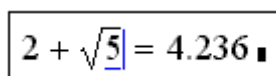
Ввод осуществляется в месте расположения крестообразного курсора. В ходе расчетов формулы и графические области обрабатываются сверху вниз, а текстовые блоки игнорируются.

Ввод формул

Чтобы набрать формулу, необходимо установить крестообразный курсор в нужное место и начать ввод букв, цифр, знаков арифметических операций. При этом создается область формулы, в которой появляется уголковый курсор, охватывающий ее текущий элемент.



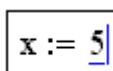
Для вывода значения формулы используется знак "=", после чего на экране появится результат.



Если в формуле используются переменные, необходимо сначала присвоить им значения, иначе на экран будет выведено сообщение об ошибке.

Присваивание переменным значений

Присваиванием называется задание переменным значения. Для присваивания значений переменным в Mathcad используется знак " := ".



Такая запись означает, что переменной x присвоено значение 5.

Знак присваивания набирается нажатием одной клавиши ":" (или с использованием панели **Вычисление (оценка)**, которая открывается при щелчке на кнопке 

В последних версиях Mathcad для присваивания переменным значения вместо двоеточия можно использовать знак равенства “=” (при этом на экран будет выведен знак присваивания “:=”), но только при первом присваивании. Основное назначение оператора “=” – это вывод значения выражения. До первого присваивания переменная не определена (не имеет значения), поэтому знак “=” срабатывает как оператор присваивания. При дальнейшем переопределении значений переменной знак присваивания должен вводиться с помощью двоеточия, так как если набрать знак “=”, на экран будет выведено текущее, ранее присвоенное значение переменной.

Различают *локальное* и *глобальное* присваивание. Если переменной присваивается значение с помощью оператора “:=”, то такое присваивание является локальным. При глобальном присваивании переменная получает заданное значение независимо от того, в каком месте документа стоит оператор глобального присваивания. Например, если переменной присвоено таким образом некоторое значение в конце документа, то она будет иметь это же значение и в начале документа. В дальнейшем значение переменной можно изменить с помощью знака локального присваивания. Глобальное присваивание осуществляется с помощью знака $\equiv$ (три горизонтальные черточки).

Приведем примеры присваиваний (идущие последовательно).

$\ln(x)=$ значение функции не может быть выведено, так как переменная x не определена

$x:=5$ переменной x присваивается значение 5;

$\ln(x)=1.609$ выводится значение функции для $x=5$

$x:=2$ переменной x присвоено новое значение 2

$\exp(x)=7.389$ выводится значение функции для $x=2$

$y=50$ переменная y доступна, поскольку определена глобально в конце документа

$\ln(y)=3.912$

$x+y=52$

$y:=123$ переменная y переопределяется локально

$$\ln(y)=4.812$$

$$x+y=125$$

$$y\equiv 50$$

переменной y глобально присвоено значение 50

Использование неопределенной переменной в каком-либо выражении является ошибкой и ведет к окраске ее имени ярко-красным цветом.

Основные приемы ввода формул

Для ввода знаков арифметических операций в формулах можно использовать панели математических инструментов. Но наиболее распространенные знаки удобно вводить с клавиатуры.

Название операции	Ввод	Изображение на экране
Присваивание	a:8 b:5	a:=8 b:=5
Вывод значения	a= b=	a=8 b=5
Сложение	a+b=	a+b=13
Вычитание	a-b=	a-b=3
Умножение	a*b=	a·b=40
Деление	a/b=	$\frac{a}{b} = 0.2$
Возведение в степень	a^2= b^3=	a ² =64 b ³ =125
Квадратный корень	\a= \b=	$\sqrt{a} = 2.828$ $\sqrt{b} = 2.236$
Абсолютное значение	-5=	-5 =5
Задание индексированной переменной	C[n	C _n

Ввод текста

Для ввода текста необходимо осуществить следующую последовательность действий:

1. Установить крестообразный курсор на место ввода.
2. В меню команд выбрать пункты **Вставка→Текстовая область**, после чего на экране появится блок для ввода текста, который имеет маркеры изменения размера в виде черных прямоугольников. В текстовом блоке курсор имеет вид красной вертикальной черты и отмечает место ввода.
3. С помощью раскрывающегося списка выбора шрифта на панели форматирования установить нужный шрифт (для ввода русскоязычного текста Arial Cyr).
4. Посимвольно набрать текст, используя типовые средства текстового редактора. Текстовый блок в процессе набора текста расширяется.
5. Для завершения ввода текста, щелкнув левой кнопкой мыши вне текстовой области.

Для коррекции текста необходимо подвести указатель мыши к месту коррекции и щелкнуть левой кнопкой. При работе с текстом можно использовать все приемы редактирования, применяющиеся при подготовке текстовой документации (например, в редакторе Word). Управление стилем текста и его расположением внутри текстового блока можно осуществлять с помощью панели форматирования.

Текстовую область можно растягивать, изменять ее размер, копировать, переносить в другое место рабочего листа, удалять и т.д..

Построение графиков

Для ввода графика функции необходимо сначала задать эту функцию. Затем поместить крестообразный курсор в место расположения левого верхнего угла графиков и выбрать нужный тип графика. Выбрать тип графика можно следующими способами:

1. Выбрав в меню **Вставка** пункт **График**.
2. Выбрав нужный тип графика на панели **Графики**.
3. С помощью клавиши **@**.

После выбора типа графика на экран будет выведен график, который можно далее форматировать в соответствии с желанием пользователя.

Рассмотрим построение простого двумерного графика $y(x) = \sin(x)$.

1. Задаем функцию $y(x)$.

$$y(x) := \sin(x)$$

2. Располагаем крестообразный курсор в том месте рабочего листа, где должен появиться график.

3. В панели **График** выбираем **X-Y-график** , после чего на экране появится шаблон графика.

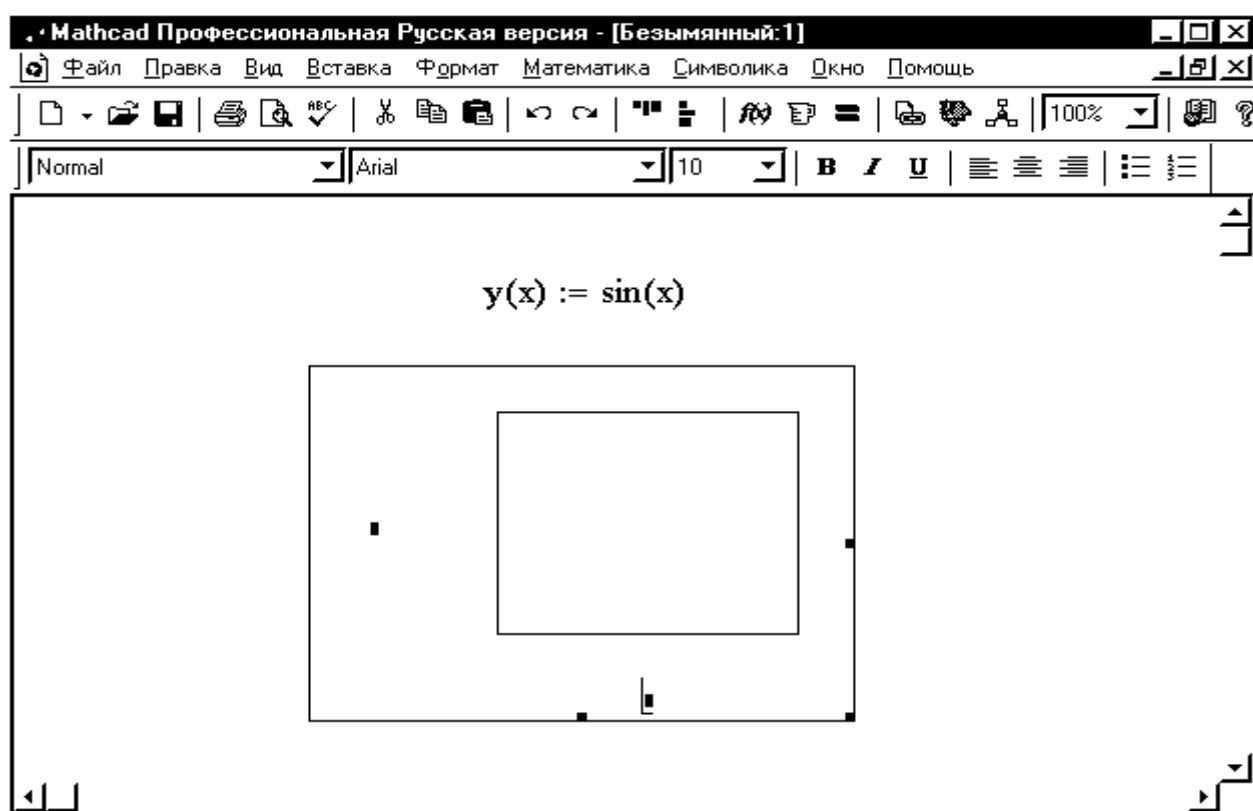


Рис. 7.4.

4. Пустое поле в середине горизонтальной оси предназначено для ввода независимой переменной. В него вводим имя переменной x . Пустое поле в середине вертикальной оси предназначено для ввода имени функции. В него вводим имя функции $y(x)$.

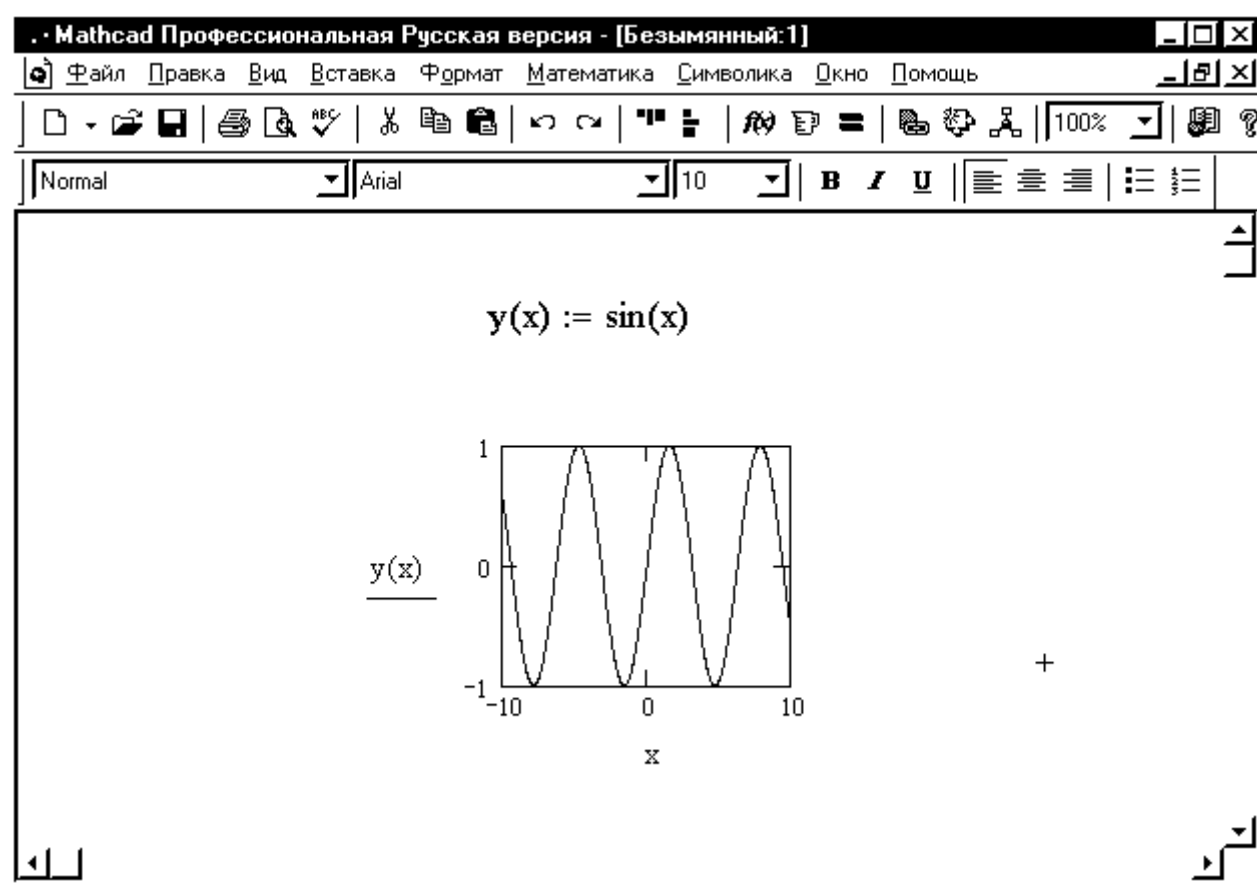


Рис. 7.5.

Mathcad всегда автоматически выбирает границы на осях координат. Для их изменения необходимо войти в область графика и исправить числа, находящиеся слева и справа от переменной x (при этом изменится масштаб по оси абсцисс), а также сверху и снизу от имени функции $y(x)$ (при этом изменится масштаб по оси ординат).

7.3. Этапы решения задач оптимизации в MATHCAD

Решение задач оптимизации средствами Mathcad осуществляется с использованием встроенных функций **Maximize** (для решения задач максимизации) и **Minimize** (для решения задач минимизации).

Функции вводятся в следующем виде:

$\text{Maximize}(f, x_1, x_2, \dots, x_n)$ или $\text{Minimize}(f, x_1, x_2, \dots, x_n)$, где f - имя целевой функции; $x_1, x_2, \dots, x_n$ – параметры.

Решение включает следующие основные этапы:

1. Ввод целевой функции (например, $f(x_1, x_2) := x_1 + x_2$)
2. Ввод начальных значений переменных (например, $x_1 := 1$
 $x_2 := 2$)
3. Ввод служебного слова Given.
4. Ввод ограничений (например, $x_1 + 2x_2 \leq 0$
 $x_1 \geq 0$
 $x_2 \geq 0$)

При вводе знаков ограничений используется панель **Логическая**.

5. Ввод встроенной функции Maximize или Minimize.
(Например, $\text{Maximize}(f, x_1, x_2) =$). При этом на экран будет выведено решение.

Фрагмент рабочего листа Mathcad для решения данной задачи:

$$\begin{aligned} f(x_1, x_2) &:= x_1 + x_2 \\ x_1 &:= 1 & x_2 &:= 2 \\ \text{Given} \\ x_1 + 2x_2 &\leq 2 \\ x_1 &\geq 0 \\ x_2 &\geq 0 \\ \text{Maximize}(f, x_1, x_2) &= \begin{pmatrix} 2 \\ 0 \end{pmatrix} \end{aligned}$$

7.4. Лабораторная работа №6

РЕШЕНИЕ ЗАДАЧ ОПТИМИЗАЦИИ СРЕДСТВАМИ MATHCAD

Цель работы: Получение практических навыков решения оптимизационных задач средствами Mathcad.

Лабораторные задания

С использованием системы Mathcad выполнить следующие задания:

1. Задание 2 из лабораторной работы №2.
2. Задание 1 из лабораторной работы №3.
3. Задание 1 из лабораторной работы №5.

Отчет по лабораторной работе должен содержать :

1. Оптимизационные модели в соответствии с вариантом.
2. Краткое описание основных этапов решения в среде Mathcad.
3. Результаты решения задачи (рабочие листы Mathcad).
4. Выводы.

ЗАКЛЮЧЕНИЕ

В учебном пособии рассматривается технология решения задач оптимизации различных классов с использованием стандартных систем EXCEL и MATHCAD. Особое внимание уделяется этапу построения и типизации оптимизационных моделей при поиске оптимальных решений. Даны приемы сведения задач оптимального выбора к математическим моделям, допускающим применение стандартных инструментальных средств. Рассматриваются теоретические и алгоритмические основы построения методов оптимизации различных классов. Использование эффективных методов и алгоритмов поиска оптимальных решений играет важную роль при проектировании, создании и эксплуатации современных САПР и информационных систем.

Материал учебного пособия может быть использован в лекционных курсах, рассматривающих вопросы принятия оптимальных решений, при проведении лабораторных и практических занятий, в курсовом и дипломном проектировании.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Акулич И. Л. Математическое программирование в примерах и задачах: учеб. пособие / И.Л. Акулич. М.: Высш. шк., 1986. – 319 с.
2. Алексеев О. Г. Комплексное применение методов дискретной оптимизации / О.Г. Алексеев. М.: Наука, 1987. – 280 с.
3. Банди Б. Основы линейного программирования / Б. Банди. М.: Радио и связь, 1989. – 86 с.
4. Батищев Д. И. Оптимизация в САПР / Д.И. Батищев, Я.Е. Львович, В.Н. Фролов. Воронеж: ВГТУ, 1997. – 416 с.
5. Белецкая С.Ю. Математические методы поиска оптимальных решений: учеб. пособие / С.Ю. Белецкая. Воронеж: ВГТУ, 2006. 200 с.
6. Васильев Ф. П. Численные методы решения экстремальных задач: учеб. пособие для вузов / Ф.П. Васильев. М.: Наука, 1988. – 552 с.
7. Мину М. Математическое программирование. Теория и алгоритмы / М. Мину. М.: Наука, 1990. – 270 с.
8. Пантелеев А. В. Методы оптимизации в примерах и задачах: учеб. пособие / А.В. Пантелеев, Т.А. Летова. М.: Высш. шк., 2002. – 544 с.
9. Реклейтис Г. Оптимизация в технике / Г. Реклейтис, А. Рейвиндран, К. Рэгсдел. В 2 кн.: пер. с англ. М.: Мир, 1986. – 346 с.
10. Таха Х. Введение в исследование операций / Х. Таха. М.: Изд-во “Вильямс”, 2001. – 912 с.
11. Штойер Р. Многокритериальная оптимизация. Теория, вычисления, приложения / Р. Штойер. М.: Радио и связь, 1992. – 504 с.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
1. МАТЕМАТИЧЕСКИЕ И АЛГОРИТМИЧЕСКИЕ ОСНОВЫ ПОИСКА ОПТИМАЛЬНЫХ РЕШЕНИЙ.....	4
1.1. Обобщенная математическая модель оптимального выбора.....	4
1.2. Классификация задач оптимизации.....	7
1.3. Алгоритмическая схема автоматизированного поиска оптимальных решений.....	8
2. РЕШЕНИЕ ЗАДАЧ ОДНОМЕРНОЙ ОПТИМИЗАЦИИ.....	12
2.1. Метод Ньютона.....	13
2.2. Метод секущих.....	15
2.3. Метод дихотомии.....	16
2.4. Метод Фибоначчи.....	18
2.5. Метод золотого сечения.....	21
2.6. Лабораторная работа № 1.....	23
3. ЛИНЕЙНАЯ ОПТИМИЗАЦИЯ.....	25
3.1. Модели линейной оптимизации.....	25
3.2. Методы решения задач линейной оптимизации.....	32
3.2.1. Симплекс-метод.....	32
3.2.2. Метод искусственного базиса.....	38
3.3. Этапы решения задач линейной оптимизации в EXCEL.....	43
3.4. Лабораторная работа № 2.....	61
4. ТРАНСПОРТНЫЕ ЗАДАЧИ ЛИНЕЙНОЙ ОПТИМИЗАЦИИ.....	63
4.1. Особенности задач транспортного типа.....	63
4.2. Решение транспортных задач методом потенциалов.....	65
4.3. Автоматизация решения задач транспортного типа средствами EXCEL.....	72
4.4. Лабораторная работа № 3.....	76
5. РЕШЕНИЕ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ.....	83
5.1. Модели дискретной оптимизации.....	83
5.2. Методы решения задач дискретной оптимизации.....	89

5.2.1. Метод отсечений Гомори.....	89
5.2.2. Метод ветвей и границ.....	95
5.3. Решение задач дискретной оптимизации с использованием EXCEL.....	102
5.4. Лабораторная работа № 4.....	106
6. НЕЛИНЕЙНАЯ ОПТИМИЗАЦИЯ.....	112
6.1. Постановка задач нелинейной оптимизации.....	112
6.2. Методы решения задач нелинейной оптимизации.....	112
6.2.1. Преобразование задач с ограничениями к задачам безусловной оптимизации.....	112
6.2.2. Принципы построения и классификация методов безусловной оптимизации.....	116
6.2.3. Методы нулевого порядка.....	117
6.2.4. Методы первого порядка.....	123
6.2.5. Методы второго порядка.....	125
6.3. Решение задач нелинейной оптимизации в EXCEL.....	126
6.4. Лабораторная работа № 5.....	130
7. РЕШЕНИЕ ЗАДАЧ ОПТИМИЗАЦИИ СРЕДСТВАМИ MATHCAD.....	133
7.1. Элементы пользовательского интерфейса MATHCAD.....	133
7.2. Основы работы в системе MATHCAD.....	144
7.3. Этапы решения задач оптимизации в MATHCAD.....	155
7.4. Лабораторная работа № 6.....	156
ЗАКЛЮЧЕНИЕ.....	157
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	158