МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение
высшего профессионального образования
«Оренбургский государственный университет»

Кафедра математических методов и моделей в экономике

Д. В. ДОМАШОВА

МЕТОДЫ РЕШЕНИЯ ЗАДАЧ МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ И САМОСТОЯТЕЛЬНОЙ РАБОТЕ СТУДЕНТОВ

Рекомендовано к изданию Редакционно-издательским советом
государственного образовательного учреждения высшего профессионального
образования «Оренбургский государственный университет»

Оренбург 2008

УДК 519.615.7(075.6)
ББК 22.19я7
Д 66

Рецензент
кандидат экономических наук, доцент К.И. Майстренко

Домашова Д.В.
Д 66 **Методы решения задач многокритериальной оптимизации [Текст]: методические указания к практическим занятиям и самостоятельной работе студентов/ Д.В. Домашова. – Оренбург: ГОУ ОГУ, 2008. – 49 с.**

Методические указания содержат описания методов решения задач многокритериальной оптимизации, а также методику использования разработанного программного средства для решения задач целевого программирования.

Методические указания предназначены студентам специальности 080116 «Математические методы в экономике» и студентам всех специальностей, изучающим дисциплины «Математические методы и модели исследования операций», «Методы оптимизации», «Теория игр и исследование операций» и другие курсы, связанные с использованием методов многокритериальной оптимизации.

ББК 22.19я7

Содержание

Введение

Почти всякая сложная задача принятия решения в технической, экономической или любой другой сфере деятельности человека является многоцелевой, т.к. при выборе наилучшего варианта приходится учитывать много различных требований (критериев), и среди этих требований часто встречаются противоречащие друг другу. Однако почти все математические методы оптимизации предназначены для нахождения экстремума одной функции - т.е. для одной цели. Поэтому пытаются свести многоцелевую задачу к одноцелевой. Эта процедура в большинстве случаев приводит к серьезному искажению существа проблемы и, следовательно, к неоправданной замене одной задачи другой.

Если при решении одноцелевых задач методологических проблем не возникает, а возможны только вычислительные трудности, то иначе обстоит дело с многоцелевыми решениями. Не существует единого мнения относительно того, что следует считать наилучшей альтернативой в задаче с несколькими целевыми функциями, которые противоречивы и достигают минимума в различных точках множества альтернатив.

Поэтому, в настоящее время разработан ряд методов, не "уходящих" от сложности проблемы и предпочитающих учитывать все ее стороны. Такие методы основаны на принципе компромисса, то есть принятия взвешенного решения, в котором фигурируют в определенной пропорции все действующие факторы. При этом в некоторых методах предлагается не однозначный ответ, а лишь область разумных (рациональных) решений. Принятие же однозначного решения остается прерогативой лица принимающего решение (ЛПР).

В данной работе будет рассмотрено решение задачи многокритериальной оптимизации с помощью интерактивного целевого программирования.

1 Методы решения задач многокритериальной оптимизации

1.1 Постановка и основные свойства задач многокритериальной оптимизации

Анализ многих реальных практических проблем, с которыми сталкивались специалисты по исследованию операций, естественным образом привел к появлению класса многокритериальных задач.

Любая операция представляет собой совокупность целенаправленных действий, и проведение практически любой операции, как правило, предполагает достижение не одной, а нескольких целей [1].

Формально, постановка задачи многокритериальной оптимизации заключается в следующем. Пусть D – произвольное множество допустимых решений (альтернатив). $f_1, \dots f_m$ - числовые функции (числовые функции, критерии), заданные на множестве D. $f(x) = (f_1(x), \dots f_m(x))$ - векторный критерий. Задача:

$$f(x) \to \min_{x \in D}, \qquad (1.1)$$

называется задачей многокритериальной оптимизации.

Идеальным в задаче (1.1) является поиск такого решения, который принадлежит пересечению множеств оптимальных решений всех задач, то есть $\bigcap_{i=1}^{m} \operatorname{Arg} \min_{x \in D} f_i(x)$ - множество абсолютных решений.

Обычно это множество пусто, так как несколько функций, вообще говоря, не достигают экстремума в одной и той же точке. Одновременное достижение минимума по всем критериям на одном решении, как правило, невозможно. Выход состоит в поиске некоторого компромисса в достижении локальных целей.

При появлении многих критериев задачи выбора наилучшего решения приобретают следующие особенности:

- задача имеет уникальный, новый характер – нет статистических данных, позволяющих обосновать соотношения между различными критериями;
- на момент принятия решения принципиально отсутствует информация, позволяющая объективно оценить возможные последствия выбора того или иного варианта решения. Но поскольку решение, так или иначе, должно быть принято, то недостаток информации необходимо восполнить. Это может быть сделано лишь людьми на основе их опыта и интуиции.

ЛПР должен сформулировать некоторый принцип компромисса и придерживаться его при выборе оптимального решения. Принцип компромисса должен определить свойства оптимального решения и дать ответ на главный вопрос: в каком смысле оптимальное решение лучше всех других решений?

Кроме того, оно должно принадлежать множеству допустимых решений задачи. Число возможных принципов компромисса очень велико. Поэтому при решении многокритериальных задач возникает ряд проблем, носящих концептуальный характер.

Основная сложность логического анализа многокритериальных задач состоит в том, что в них появляется эффект несравнимости исходов. Выбор между несравнимыми исходами является сложной концептуальной проблемой и составляет основное содержание многокритериальной оптимизации.

Основное отношение, по которому производится сравнение исходов – это отношение доминирования по Парето, которое определяется следующим образом.

Пусть $x^1 = (x_1^1,...,x_n^1)$, $x^2 = (x_1^2,...,x_n^2)$- некоторые допустимые решения. Пусть Z_j - множество значений функций $z_j = f_j(x_1,...,x_n)$, $j=\overline{1,m}$, то есть множество всех оценок по j-му критерию. Тогда множество Z, состоящее из всевозможных упорядоченных наборов оценок по критериям f_j, $j=\overline{1,m}$, называется множеством критериальных векторов. Любой элемент $z \in Z$ представляет собой вектор $z = (z_1,...,z_m)$, где $z_j \in Z_j$. Критериальный вектор исхода содержит полную информацию о ценности (полезности) этого исхода для принимающего решение. Сравнение двух исходов заменяется сравнением их критериальных векторов x^1 и x^2.

Критериальный вектор $z^1 = (z_1^1, z_2^1,...,z_m^1)$ доминирует по Парето критериальный вектор $z^2 = (z_1^2, z_2^2,...,z_m^2)$, если для всех $j=\overline{1,m}$ выполняется неравенство $z_j^1 \le z_j^2$, причем по крайне мере для одного индекса $j=\overline{1,m}$ неравенство должно быть строгим.

Пусть $Q \subseteq Z$ - некоторое множество критериальных векторов. Критериальный вектор $z^* \in Q$ называется Парето-оптимальным в Q, если он не доминируется по Парето никаким другим критериальным вектором. То есть если в множестве Q не существует такого критериального вектора z, который доминирует по Парето критериальный векор z^*.

Перенесем эти понятия на исходы.

Исход x^1 доминирует по Парето исход x^2 (записывается $x^1 \overset{par}{>} x^2$), если векторная оценка исхода x^1 доминирует по Парето векторную оценку исхода x^2. Содержательно это означает, что исход x^1 не хуже, чем x^2 по любому из рассматриваемых критериев, причем, по крайней мере, по одному из этих критериев x^1 лучше, чем x^2. Поэтому при сравнении двух исходов x^1, $x^2 \in D$, для которых $x^1 \overset{par}{>} x^2$, принимающий решение безусловно отдаст предпочтение исходу x^1.

Исход $x^* \in D$ называется Парето-оптимальным исходом в множестве D, если он не доминируется по Парето никаким другим исходом их множества D. Парето-оптимальность исхода x^* означает, что он не может быть улучшен ни по одному из критериев без ухудшения по какому-нибудь другому критерию.

Перейдем теперь к основной проблеме - проблеме оптимальности для многокритериальных ЗПР. Определим вначале необходимое условие оптимальности: если исход $x \in D$ не является Парето оптимальным, то он не может «претендовать на роль» оптимального решения. Действительно, в этом случае существует такой допустимый исход $x' \in D$, что $x' \overset{Par}{>} x$; тогда, как отмечалось выше принимающий, решение предпочтет исход x' исходу x. Значит исход x не оптимален.

Итак, «кандидатом» на оптимальное решение многокритериальной ЗПР может являться только Парето-оптимальный исход. Однако в типичных случаях Парето-оптимальных исходов может быть бесчисленное множество. Дать однозначный ответ на вопрос, какой же из Парето-оптимальных исходов следует считать оптимальным, для общего случая, не имея дополнительной информации о критериях, невозможно, так как любые два Парето-оптимальных исхода не сравнимы относительно доминирования по Парето.

Для решения задач многокритериального выбора предложено множество математических методов. Методы прикладной теории принятия решений различаются способом представления и обработки экспертных знаний. Подход к проблеме выбора может основываться на отношениях порядка среди альтернатив (классическая модель принятия решений, в которой каждой альтернативе ставится в соответствие некоторое число) или на отношениях включения (поведенческая модель, основанная на принадлежности альтернатив к некоторому множеству).

Одними из методов решения задач многокритериальной оптимизации являются методы теории полезности.

1.2 Анализ методов решения задач многокритериальной оптимизации

Теория многомерной полезности позволяет для задач получить функцию многомерной полезности, максимальное значение которой соответствует наиболее предпочтительному варианту. Многомерная функция полезности обычно получается как аддитивная или мультипликативная комбинация одномерных функций, которые строятся на основании опроса экспертов и позволяют провести ранжирование возможных исходов без взаимного сравнения альтернатив. При этом делается допущение о взаимной независимости критериев по полезности. Процедура построения функции полезности требует привлечения значительных объемов информации и является достаточно трудоемкой. Достоинством этого подхода является возможность оценки любого количества альтернативных вариантов с использованием полученной функции. В случае неустойчивой исходной информации применение методов теории полезности становится малоэффективным.

Существует два основных недостатка, связанные с данным подходом. Во-первых, предполагается (неявно), что человек может делать точные количественные измерения. Это далеко не так. Например, психологические исследования показали, что нет надежного способа количественного измерения весов критериев. Во-вторых, этот подход требует от ЛПР «немедленного» назначения всех основных параметров, не давая ему возможности провести исследования проблемы привычным для человека методом «проб и ошибок».

Математически задача формулируется следующим образом. Пусть $U(z)$ - функция полезности ЛПР. Тогда идеальным решением линейной многокритериальной задачи было бы отыскание решения задачи:

$$\begin{aligned} &U(z) \to \max \\ &z = C \cdot x, \quad x \in D, \end{aligned} \tag{1.2}$$

Если x^0 - решение задачи (1.2) и z^0 - соответствующий полученному решению оптимальный критериальный вектор, то решение x^0 - оптимальное решение задачи (1.1).

Предполагается, что ЛПР изначально имеет свои предпочтения на некотором подмножестве $X \subset \{x : x \in D\}$. Это означает, что для каждой пары $x_1 \in D, x_2 \in D$ имеет место одно из трех соотношений:

$x_1 \succ x_2$ - набор x_1 предпочтительнее x_2;

$x_1 \prec x_2$ - набор x_1 менее предпочтителен, чем x_2;

$x_1 \sim x_2$ - для ЛПР оба набора обладают одинаковой степенью предпочтения.

Отношения предпочтения обладают по крайней мере следующими свойствами:

1) если $x_1 \succ x_2, x_2 \succ x_3$, то $x_1 \succ x_3$ (транзитивность);
2) если $x_1 > x_2$, то $x_1 \succ x_2$ (ненасыщаемость).

Отношения предпочтения каждого ЛПР при определенных слабых предположениях, касающихся предпочтений, можно представить в форме индикатора предпочтений, то есть такой функции полезности $u(z)$, что из $x_1 \succ x_2$ следует $u(z_1) > u(z_2)$ и из $x_1 \sim x_2$ следует $u(z_1) = u(z_2)$. Для каждого ЛПР такое представление многовариантно.

Предполагается, что функция полезности обладает следующими свойствами:

1) $\frac{\partial u}{\partial x_i} > 0$ - с ростом потребления блага полезность растет;
2) $\lim\limits_{x_i \to 0} \frac{\partial u}{\partial x_i} = \infty$ - небольшой прирост блага при его первоначальном отсутствии резко увеличивает полезность;
3) $\frac{\partial^2 u}{\partial x_i^2} < 0$ - с ростом потребления блага скорость роста полезности замедляется;
4) $\lim\limits_{x_i \to 0} \frac{\partial u}{\partial x_i} = \infty$ - при очень большом объеме блага его дальнейшее увеличение не приводит к увеличению полезности.

Как правило, функция полезности вогнутая функция. Ее линия уровня выпуклая и убывающая.

Для функции полезности справедливы следующие две теоремы.

Пусть $U: R^m \to R$ - покоординатно возрастающая функция полезности. Если $z^0 \in z$ - оптимальная точка, то критериальный вектор z^0 - не доминируем. ($z^0 = \arg\max_{z \in Z} U(z)$).

Пусть $\bar{z} \in Z$ - не доминируется. Следовательно, существует покоординатно возрастающая функция полезности. $U: R^m \to R: \bar{z} = \arg\max_{z \in Z} U(z)$.

Как уже было отмечено выше, главная проблема этого подхода – построение функции полезности. Но этот подход полезен в концептуальном плане. Например, мы получаем возможность дать определения оптимального решения задачи многокритериальной оптимизации.

Еще одним методом решения задач многокритериальной оптимизации является сужение Парето-оптимальных исходов (в идеале до одного элемента) с помощью некоторых формализованных процедур. Такое сужение может быть произведено только при наличии дополнительной информации о критериях или о свойствах оптимального решения.

Будем считать, что многокритериальная задача принятия решений (ЗПР) задана в виде $\langle D; f_1, \ldots f_m \rangle$, где $f_j \quad (j = \overline{1,m})$ - негативные критерии. Рассмотрим некоторые способы сужения Парето-оптимального множества.

Дополнительная информация об оптимальном исходе $f^* \in D$ в этом случае имеет следующий вид:

$$f_j(x^*) \le f_{j0}, \quad j = \overline{1,m}, \tag{1.3}$$

Число f_{jo} рассматривается здесь как нижняя граница по j-му критерию.

Указание нижних границ критериев по критериям $j = \overline{1,m}$ не может быть «извлечено» из математической модели ЗПР; набор оценок $(f_{10}, \ldots f_{m0})$ представляет собой дополнительную информацию, полученную от принимающего решение.

При указании нижних границ критериев оптимальным может считаться только такой Парето-оптимальный исход, для которого оценка по каждому из критериев $j = \overline{1,m}$ не ниже назначенной оценки f_{jo}. Таким образом, происходит сужение Парето-оптимального множества за счет условия (1.3.1). Ясно, что при увеличении значений $f_{jo} \ (j = \overline{1,m})$ Парето-оптимальное множество «сокращается».

Окончательный выбор Парето-оптимального решение производится из суженного Парето-оптимального с помощью метода субоптимизации.

Субоптимизацию производят следующим образом: выделяют один из критериев, а по всем остальным критериям назначают нижние границы. Оптимальным при этом считается исход, максимизирующий выделенный критерий на множестве исходов, оценки которых по остальным критериям не ниже назначенных.

Пусть среди частных критериев $f_j(x), \ j = \overline{1,m}$ $f_1(x)$ является основным. Тогда задачу (1.3.1) сводят к однокритериальной задаче, переводя остальные критерии в ограничения.

$$\begin{aligned} & f_1(x) \to \min \\ & f_k(x) \le f_{k0}, \quad k = \overline{2,m}, \\ & x \in D \end{aligned} \tag{1.4}$$

С помощью метода субоптимизации задача многокритериальной оптимизации превращается в задачу скалярной оптимизации на суженном допустимом множестве. Выделение одного из критериев, а также указание нижних границ для остальных критериев основано на дополнительной информации, получаемой от принимающего решение. Следовательно, окончательное решение здесь также имеет субъективный характер.

Главными вопросами, возникающими при применении данного подхода являются выбор верхних границ f_{k0} и какой критерий выбрать главным.

Основной недостаток метода состоит в том, что оптимальное решение становится здесь субъективным, так как зависит, во-первых, от величин назначаемых нижних границ критериев и, во-вторых, от выбора основного критерия.

Ранжирование критериев (лексикографическая оптимизация) основан на упорядочении критериев по их относительной важности. После этого процедуру нахождения оптимального решения проводят следующим образом. На первом шаге отбирают исходы, которые имеют максимальную оценку по важнейшему критерию. Если такой исход единственный, то его и считают оптимальным. Если же таких исходов несколько, то среди них выбирают те, которые имеют максимальную оценку по следующему важнейшему критерию и т.д. В результате такой процедуры всегда остается (по крайней мере, в случае конечного множества исходов) единственный исход – он и будет оптимальным.

Пусть критерии упорядочены в соответствии с их значимостью: номер целевой функции отражает ранг соответствующего критерия. Пусть D^* - множество Парето-оптимальных решений. Процедура выбора единственного эффективного решения из D^* начинается с критерия 1-го ранга.

1) ищем $D^{1*} = \arg\min\limits_{x \in D^*} f_1(x)$ - множество всех допустимых решений из D^*, которое минимизирует $f_1(x)$.

2) ищем $D^{2*} = \arg\min\limits_{x \in D^{1*}} f_2(x)$

…

m) ищем $D^{m*} = \arg\min\limits_{x \in D^{m-1*}} f_m(x)$

Основными недостатками метода лексикографической оптимизации являются следующие.

– для реализуемости предложенного метода подмножеству $D^{m-1*} \subset D$ должно соответствовать подмножество множества Парето, состоящее

более чем из одного элемента, что в общем случае редко выполняется;
- при практическом применении данного метода возникают содержательные трудности в установлении полной упорядоченности критериев по их относительной важности;
- фактически при использовании этого метода принимается во внимание только первый – важнейший критерий. Например, следующий за ним по важности критерий учитывается только тогда, важнейший критерий достигает максимума на нескольких исходах.

С помощью метода уступок частично преодолевается недостаток метода ранжирования критериев.

В данном методе осуществляется поиск не единственного точного оптимума, а некоторой области решений, близких к оптимальному, – квазиоптимального множества. При этом уровень допустимого отклонения от точного оптимума определяется с учетом точности постановки задачи (например, в зависимости от точности вычисления величины критериев), а также некоторых практических соображений (например, требований точности решения задачи). Вначале производится качественный анализ относительной важности критериев; на основании такого анализа критерии располагаются и нумеруются в порядке убывания важности. Такой подход позволяет значительно сузить первоначальную допустимую область D, когда переходим к следующему по важности критерию.

Алгоритм решения по методу уступок включает следующие шаги:

1) расположить критерии $f_1(x),\ldots,f_k(x)$ по их значимости (наиболее важный с точки зрения ЛПР располагается первым);
2) найти оптимальное значение f_1^* целевой функции f_1;
3) сделать уступку по первому показателю эффективности, т.е. ухудшить величину f_1^* до значения $f_1^* = f_1^* + \Delta_1$;
4) ввести в задачу дополнительное ограничение $f_1 \le f_1^{**}$;
5) найти оптимальное значение f_2^* целевой функции f_2;
6) сделать уступку по второму показателю эффективности, т.е. ухудшить величину f_2^* до значения $f_2^{**} = f_2^* + \Delta_2$;
7) ввести в задачу дополнительное ограничение $f_2 \le f_2^{**}$;
8) новую задачу с двумя дополнительными ограничениями решить по третьему показателю эффективности и т.д.

Процесс решения задачи заканчивается, когда решение будет получено по всем показателям.

Метод последовательных уступок целесообразно применять для решения тех многокритериальных задач, в которых все частные критерии естественным образом упорядочены по степени важности. Причем каждый критерий настолько существенно более важен, чем последующий, что можно ограничиться учетом только попарной связи критериев и выбирать допустимое снижение очередного критерия с учетом поведения лишь одного следующего критерия.

Признанным недостатком метода уступок является сложность подбора подходящих уступок, их выбор требует очень тщательного подхода. При задании слишком малых значений уступок возможна такая ситуация, что оптимизация по менее значимым критериям может быть вовсе не проведена, что не всегда будет устраивать ЛПР.

В большинстве случаев множество эффективных точек оказывается весьма обширным, что затрудняет выбор конкретного решения и требует введения некоторых вторичных принципов оптимальности. В силу этого при решении многокритериальных задач необходим дополнительный перебор эффективных решений. Смысл его сводится к тому, что необходимо сформулировать некоторый критерий (или их последовательность) и оптимизировать его на переговорном множестве. В математическом плане эта задача в общем случае является задачей невыпуклого программирования, так как переговорное множество, как правило, является невыпуклым.

Поэтому, вместо того, чтобы перебирать точки переговорного множества в соответствии с некоторым критерием, обычно стремятся частные критерии $f_k(x)$ свернуть в один обобщенный скалярный критерий, минимизация которого приводит к эффективному решению. В некоторых случаях вместо одного обобщенного критерия и решения одной задачи скалярной оптимизации предлагается рассмотреть последовательность обобщенных критериев и последовательность задач скалярной оптимизации. Но не все такие процедуры приводят к эффективным решениям. Рассмотрим некоторые распространенные приемы сведения задач векторной оптимизации к задачам скалярной оптимизации.

Под построением обобщенного критерия в многокритериальной задаче принятия решений понимается процедура, которая «синтезирует» набор оценок по заданным критериям (называемым в этом случае частными или локальными критериями), в единую численную оценку, выражающую итоговую полезность этого набора оценок для принимающего решение. Формально обобщенный критерий для ЗПР вида $\langle D; f_1,\ldots,f_m\rangle$ представляет собой функцию $\varphi : Z_1 \times \ldots \times Z_m \to R$, где Z_j - множество оценок по j-му критерию. Если обобщенный критерий φ построен, то для каждого допустимого исхода $x \in D$ может быть найдена численная оценка его полезности (ценности, эффективности): $f(x) = \varphi(f_1(x),\ldots,f_m(x))$. Таким образом, задание обобщенного критерия сводит задачу многокритериальной оптимизации с целевой функцией f. Наиболее распространенным обобщенным критерием является «взвешенная сумма частных критериев», которая превращает критериальный вектор $z = (z_1,\ldots,z_m)$ в скалярную оценку

$$\varphi(z) = \alpha_1 z_1 + \ldots + \alpha_m z_m, \qquad (1.5)$$

где $\alpha_j \geq 0,\ j = \overline{1,m}$ (иногда дополнительно требуют выполнение условия $\sum_{j=1}^{m} \alpha_j = 1$). Числа α_j в этом случае называют весовыми коэффициентами. При этом весовой коэффициент α_j интерпретируется как «показатель относительной важности» j-го критерия: чем больше α_j, тем больший «вклад» дает оценка по j-му критерию в итоговую оценку $\varphi(z)$.

Справедливо следующее правило. Пусть $Q \subseteq Z$ - произвольное множество векторных оценок. Если векторная оценка $z^{*'} = (z_1^*, z_2^*, ..., z_m^*)$ доставляет максимум функции $\varphi(z) = \sum_{j=1}^{m} \alpha_j z_j$, где все $\alpha_j > 0$, то векторная оценка z^* является Парето-оптимальной в множестве Q.

Обратное утверждение справедливо не всегда, но в случае, когда множество Q является выпуклым, обратное утверждение также имеет место. Точнее, справедливо следующее правило.

Пусть $Q \subseteq Z$ - выпуклое множество, $z^* \in Q$ - Парето-оптимальная векторная оценка на множестве Q. Тогда найдутся такие неотрицательные числа $\alpha_j \geq 0,\ j = \overline{1,m}$, что функция $\varphi(z) = \sum_{j=1}^{m} \alpha_j z_j$ достигает максимума на множестве Q в точке z^*.

В общем случае второе правило гарантирует существование неотрицательных коэффициентов $\alpha_j,\ j = \overline{1,m}$, при которых максимум линейной функции $\varphi(z) = \sum_{j=1}^{m} \alpha_j z_j$ достигается в заданной Парето-оптимальной точке; существование требуемых положительных коэффициентов α_j имеет место не всегда.

Оба правила указывают «способ перебора» Парето-оптимальных точек заданного множества Q: зафиксировав положительный «вектор весов» $\alpha = (\alpha_1, ..., \alpha_m)$ и найдя максимум взвешенно суммы $\sum_{j=1}^{m} \alpha_j z_j$, получаем некоторую точку Парето-оптимального множества; в случае выпуклого множества Q все Парето-оптимальные точки множества могут быть получены таким способом при некоторых $\alpha_j \geq 0,\ j = \overline{1,m}$. [4]

Существуют следующие подходы к построению обобщенного скалярного критерия.

1. Задача минимизации скалярного критерия.

$$F(x) = \sum_{j=1}^{m} \alpha_j \cdot f_j(x) \to \min, \qquad (1.6)$$

где $\alpha_j > 0, \quad j = \overline{1,m}$.

Если $f_j(x) \geq 0$, то решение этой задачи дает эффективное решение.

2. Минимаксный обобщенный критерий.

$$F(x) = \max_{1 \leq j \leq m} \alpha_j \cdot f_j(x) \to \min, \qquad (1.7)$$

где $\alpha_j > 0, \quad j = \overline{1,m}$.

λ_k - коэффициент важности k-го решения. Точки минимизации этого критерия – эффективные решения.

3. Минимизация обобщенного скалярного критерия.

$$F(x) = \sum_{j=1}^{m} \alpha_j \frac{f_j(x) - f_{j\min}}{f_{j\min}} \to \min, \qquad (1.8)$$

где $f_{j\min}$, $j = \overline{1,m}$ - минимальное значение j-го частного критерия на области D.

Эта функция учитывает и требование нормализации критериев, так как вместо абсолютных значений скалярных критериев рассматриваются безразмерные величины их относительных отклонений от минимальных значений. Можно потребовать $\sum_{j=1}^{m} \alpha_j = 1, \alpha_j \in [0,1]$.

Основной проблемой данного метода является подбор величин α_j.

Еще одним методом решения задач многокритериальной оптимизации является целевое программирование.

2 Целевое программирование

2.1 Общая постановка задачи

Идея целевого программирования заключается в том, чтобы установить некоторый уровень достижения целей по каждому критерию. Целевое программирование (ЦП) – это идеальный подход, когда достижение критериями целевых (или пороговых) значений является особенно важным.

Особенности целевого программирования:

- любой критерий трактуется как цель;
- достижению отдельных целей приписываются приоритеты;
- вводятся переменные d_i^+, d_i^-, являющиеся мерой отклонения от целевых или пороговых уровней сверху и снизу соответственно;
- минимизируется взвешенная сумма переменных отклонений с целью найти решения, наилучшим образом удовлетворяющих целям.

Обычно точка, удовлетворяющая сразу всем целям, принадлежит утопическому множеству (множество критериальных векторов, которые одновременно удовлетворяют всем целям). Поэтому стараемся найти допустимую точку, которая достигает всех целей «как можно лучше». Методы отыскания этих точек и составляет предмет целевого программирования.

В задачах целевого программирования встречаются четыре типа целевых критерия:

а) целевая функция должна быть больше заданного порогового значения t_1: $z^1 = c^1 \cdot x \to \min, z_1 \geq t_1$;

б) целевая функция не должна превышать заданного порогового значения t_2: $z^2 = c^2 \cdot x \to \min, z_2 \leq t_2$;

в) целевая функция должна совпадать с заданным пороговым значением t_3: $z^3 = c^3 \cdot x \to \min, z_3 = t_3$;

г) значение целевой функции должно лежать в заданном интервале $[t_4^d, t_4^u]$: $z^4 = c^4 \cdot x \to \min, z_4 \in [t_4^d, t_4^u]$.

при $x \in D \in R^n$.

Соответствующие функции полезности ЛПР для всех типов целей имеют вид представленный на рисунке 1.

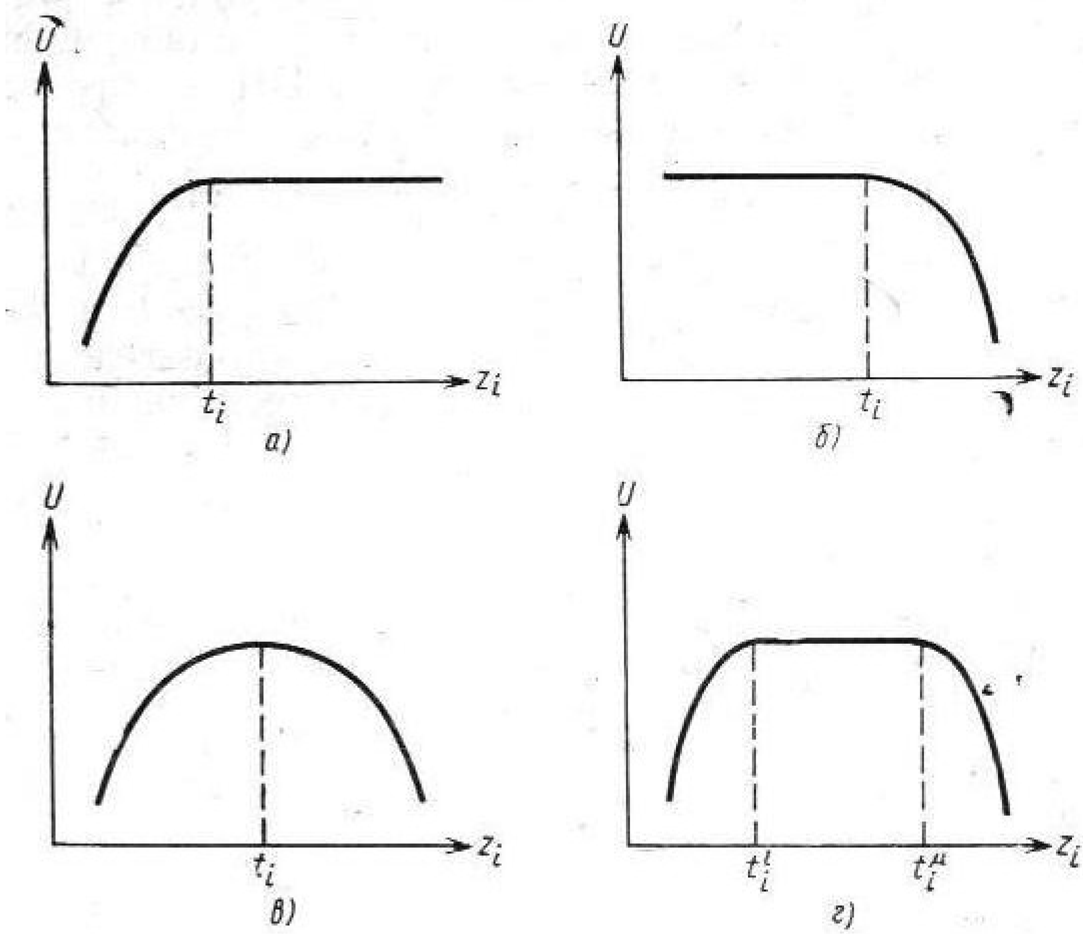


Рисунок 1. Типы функций полезности для целевых критериев.

Числа t_i - это те значения целей, выше которых, ниже которых, в которых или в диапазоне которых мы хотим находиться.

В ЦП рассматриваются два основных подхода к решению задач: архимедова модель и модель с приоритетами. Если мы имеем дело с архимедовой моделью, то точки – кандидаты в решения – генерируются путем определения тех точек из S, критериальные вектора которых являются ближайшими (в смысле взвешенной метрики L_1) к утопическому множеству в пространстве критериев. Расстояние по метрике L_1 между векторами x^k и x^l определяется следующим образом: $d(x^k, x^l)_{L1} = \sum_{i=1}^{n} |x_i^k - x_i^l|$.

Что касается модели с приоритетами, то мы генерируем решения, для которых критериальные вектора оказываются наиболее соответствующими в лексикографическом смысле точкам утопического множества.

Данная модель называется архимедовой моделью задачи ЦП:

$$w_1^- \cdot d_1^- + w_2^+ \cdot d_2^+ \cdot + w_3^+ \cdot d_3^+ + w_3^- \cdot d_3^- + w_4^- \cdot d_4^- + w_4^+ \cdot d_4^+ \to \min \qquad (2.1)$$

Целевые ограничения:

$$\begin{gathered} c^1 \cdot x + d_1^- \geq t_1 \\ c^2 \cdot x + d_2^+ \leq t_2 \\ c^3 \cdot x - d_3^+ + d_3^- = t_2 \\ c^4 \cdot x + d_4^- \geq t_4^d \\ c^4 \cdot x - d_4^+ \leq t_4^u \end{gathered} \tag{2.2}$$

¿

{¿{¿{¿{¿¿¿

¿

Ограничения исходной задачи:

$$x \in D \tag{2.3}$$

Требование неотрицательности переменных:

$$d_1^-, d_2^+, d_3^+, d_3^-, d_4^+, d_4^- \geq 0 \tag{2.4}$$

Следует отметить, что:

– целевая функция представляет собой взвешенную сумму переменных нежелательных отклонений;
– каждая цель порождает одно целевое ограничение, кроме случая диапазона, когда возникает два целевых ограничения;
– коэффициенты w_i^+, w_i^- в целевой функции представляют собой положительные штрафные веса.

Целевые ограничения являются нежесткими в том смысле, что они не ограничивают исходную область S в пространстве решений. В действительности они пополняют область допустимых решений, переводя S в пространство большей размерности и создавая таким образом расширенную (или архимедову) область допустимых решений для задачи ЦП.

Переменные w позволяют штрафовать нежелательные отклонения от цели с разной степенью жесткости.

Множество, в котором одновременно достигаются все цели, называется утопическим множеством.

Будем изображать многокритериальную задачу в пространстве решений или в пространстве критериев. Если используется пространство критериев, то демонстрируется достижимое множество в пространстве критериев Z, которое для линейной задачи многокритериальной оптимизации определяется следующим образом.

Пусть S – допустимая область в пространстве решений.

Множество $Z = \{ z \in R^k \mid z = c \cdot x, x \in S \}$ называется достижимым множеством, то есть Z – множество образов всех точек из S, где к – количество критериев.

Приведем пример.

Пример 1.

$$\text{Цель } 1: \ z_1 = x_1 + \frac{1}{2} \cdot x_2$$

$$\text{Цель } 2: \ z_2 = \frac{1}{2} \cdot x_1 + x_2$$

Требования:
1. по первой цели $z_1 \geq t_1 = 4$
2. по второй цели $z_2 \in [4.5; 6.5]$
Ограничения:
$x_1 + x_2 \leq 5$
$x_1 \leq 4$
$x_{1,2} \geq 0$
¿
{¿{¿¿¿
¿

На рисунке 2 представлена геометрическая интерпретация задачи в пространстве решений, где S – множество допустимых решений.

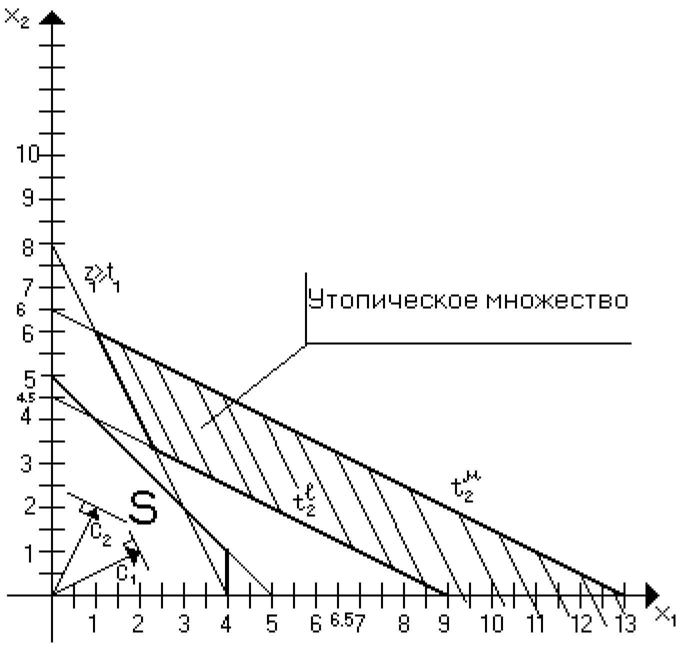


Рисунок 2. Утопическое множество в пространстве решений, в котором одновременно достигаются все цели

Представим задачу в пространстве критериев z_1 и z_2, $f^1(x^1)$, $f^2(x^2)$.
Найдем образы точек:

$(4,\ 1) \Rightarrow$

$$4 + 1 \cdot \frac{1}{2} = 4.5$$

$$4 \cdot \frac{1}{2} + 1 = 3$$

¿{¿}¿{}$\Rightarrow z^1 = (4.5,\ 3)$¿

$$(0, 0.5) \Rightarrow \begin{cases} 0 + \frac{1}{2} \cdot 5 = 2.5 \\ \frac{1}{2} \cdot 0 + 5 = 5 \end{cases} \Rightarrow z^2 = (2.5, 5)$$

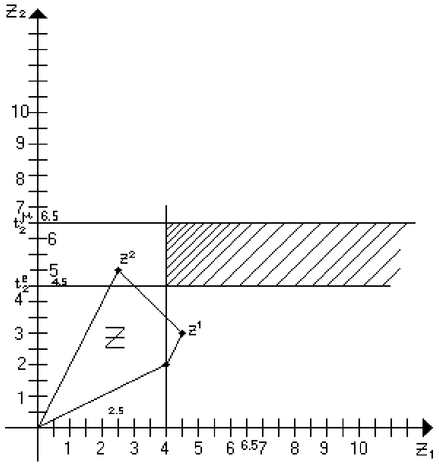


Рисунок 3. Утопическое множество в пространстве критериев

Z – множество допустимых решений в пространстве критериев.

Так как нет точек, которые были бы одновременно и допустимыми и удовлетворяли всем целям, то смысл ЦП – попытаться найти такую точку из S, критериальный вектор которой «самый лучший» по сравнению с утопическим множеством в пространстве критериев.

2.2 Линии уровня архимедовых целевых функций

Так как архимедова целевая функция представляет собой взвешенную сумму переменных нежелательных отклонений, то архимедова задача ЦП относится к классу задач оптимизации в метрике L_1. Изучим линии уровня целевых функций в архимедовых задачах ЦП.

Рассмотрим следующий пример.

Пример 2.

Цель: $z_1 = x_1$

Цель: $z_2 = x_2$

Требования:

1. по первой цели $z_1 = t_1 = 8$
2. по второй цели $z_2 = t_2 = 7$

Ограничения:

$$\begin{cases} -x_1 + 2 \cdot x_2 \le 8 \\ x_1 + x_2 \le 10 \\ x_{1,2} \ge 0 \end{cases}$$

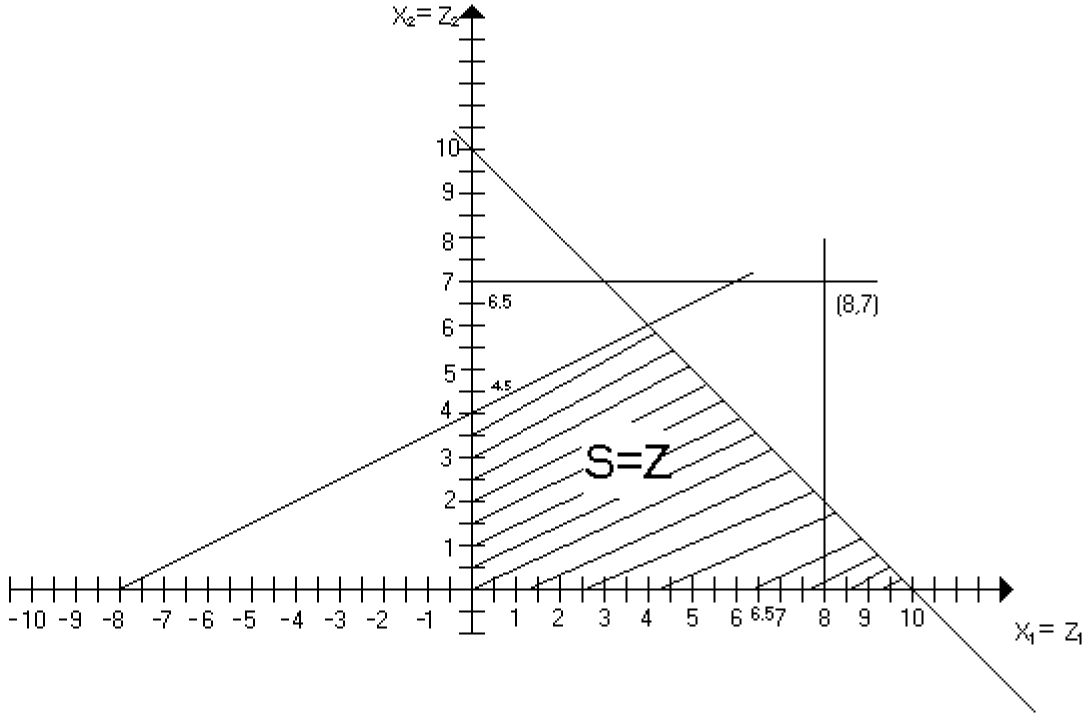


Рисунок 4. Утопическое множество в пространстве критериев – т .(8,7) (совпадает с пр-вом перем.)

Утопическое множество в пространстве критериев – это точка (8,7).

Архимедова модель данной задачи имеет вид:

$$\left\{ w_1^+ \cdot d_1^+ + w_1^- d_1^- + w_2^+ d_2^+ + w_2^- d_2^- \right\} \to \min$$

При

$$x_1 + d_1^- - d_1^+ = 8$$

$$x_2 + d_2^- - d_2^+ = 10$$

1. Пусть $w_1^+ = w_1^-$, $w_2^+ = w_2^-$ и $w_1^+ > w_2^+$. Линии уровня целевой функции в данном случае представляют собой горизонтально сжатые ромбы с центром в т. (8,7), т.о. вектор x_1=(8,2) – решение задачи, так как данная точка есть общая точка наименьшего ромба с центром в т. (8,7) и множества Z в пространстве решений (смотри рисунок 5).

Оптимальный критериальный вектор z^1 совпадает с решением задачи x^1.

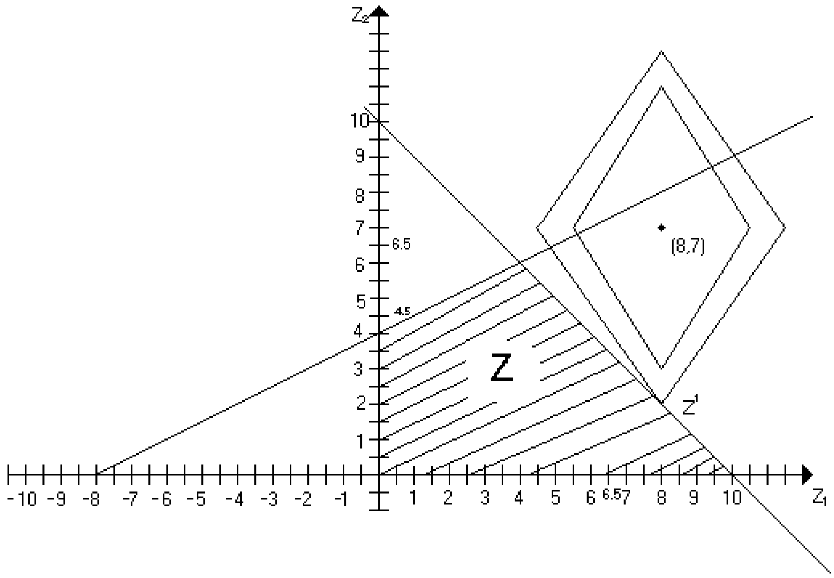


Рисунок 5. Достижение общей точки наименьшего ромба с центром в т. (8,7) и множества Z в пространстве критериев

2. Пусть $w_1^+=w_1^-$, $w_2^+=w_2^-$ и $w_1^+<w_2^+$. Линии уровня целевой функции представляют собой сжатые по вертикали ромбы, а точка $x^2=(4,6)$ –решение архимедовой задачи (смотри рисунок 6).

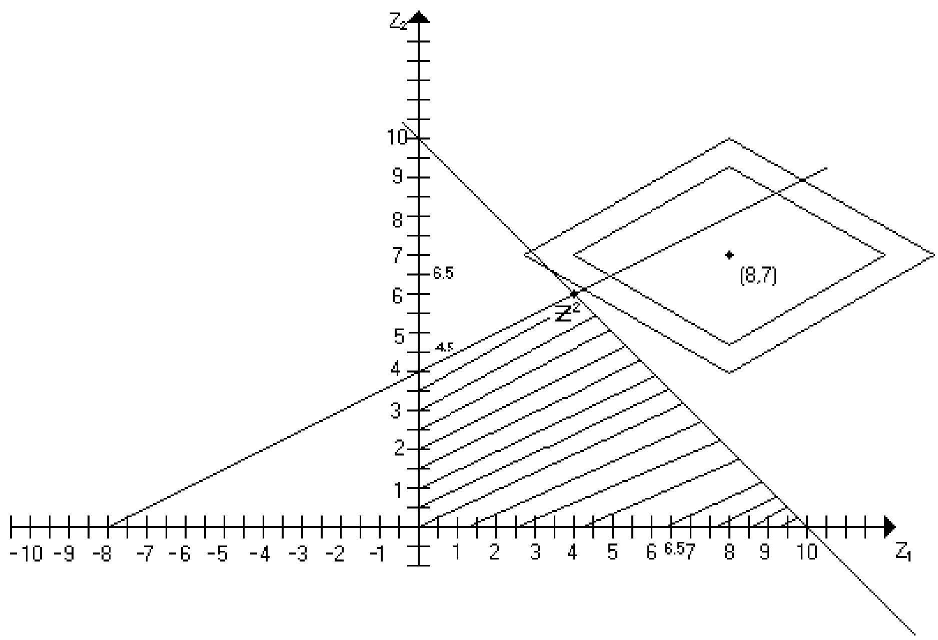


Рисунок 6. Достижение решения архимедовой задачи

Рассмотрим линии уровня целевых функций архимедовых моделей, которые определяются:

1) двумя критериями, ограниченными по диапазону.

В случае одинаковых весов линии уровня будут иметь вид как на рисунке 7:

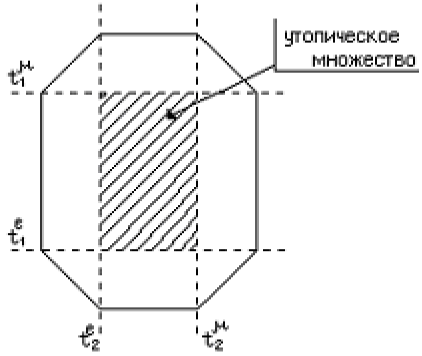


Рисунок 7. Вид линий уровня архимедовой целевой функции

2) одним критерием с требованием вида “≥”, а другим - с требованием «=». В случае одинаковых весов линии уровня будут принимать вид (рисунок 8),

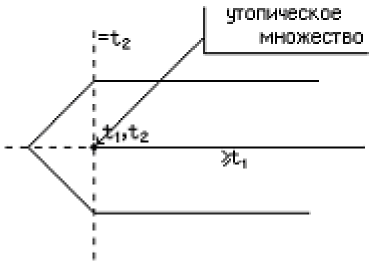


Рисунок 8. Вид линий уровня архимедовой целевой функции

3) двумя критериями с требованием вида “≥”.

В случае одинаковых весов линии уровня будут принимать вид, представленный на рисунке 9.

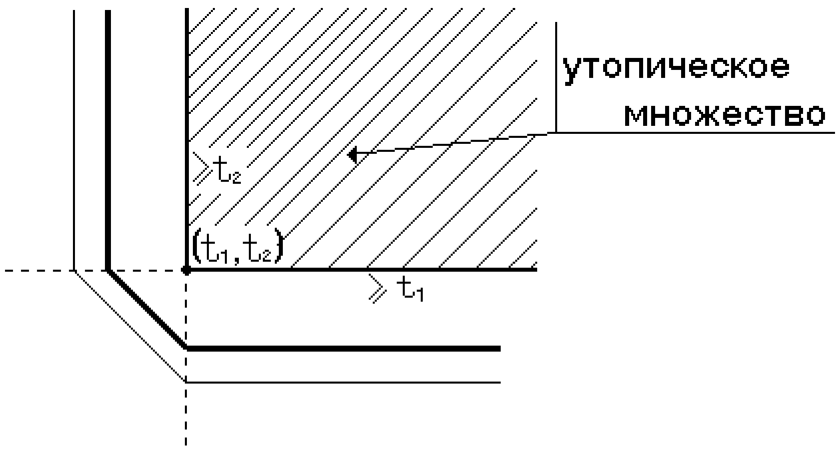


Рисунок 9. Вид линий уровня архимедовой целевой функции

Следует учитывать, что применяя методы решения задач обычных ЛП к решению задач ЦП, мы можем получать только крайние точки дополнительной области в пространстве решений для архимедовых задач ЦП, то есть крайние точки области S после её усечения целевыми ограничениями.

Приведем пример.

Пример 3.

Цель1: $z_1 = x_1$

Цель 2: $z_2 = x_2$

Требования:

1. по первой цели $z_1 \le t_1$
2. по второй цели $z_2 \ge t_2$

Архимедов критерий имеет вид:

$$w_1^+ d_1^+ + w_2^- d_2^- \to \min$$

В зависимости от весов решением в данном случае может являться:

1) либо x^1
2) либо x^2
3) либо $[x^1, x^2]$ (смотри рисунок 10)

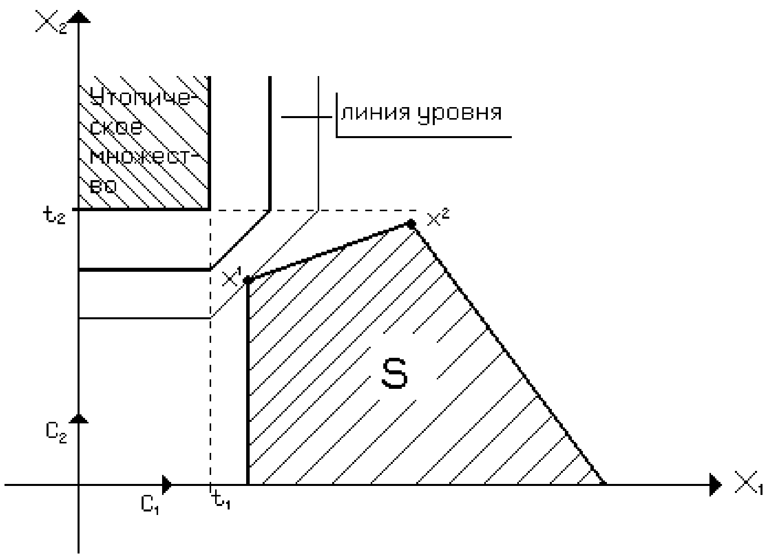


Рисунок 10. Иллюстрация к примеру 3

Рассмотрим следующий пример.

Пример 4.

Рассмотрим решение задачи ЦП, геометрическая интерпретация которой приведена на рисунке 11.

Множество решений образуют все точки множества $\wp(x^1, x^2, x^3, x^4)$, причём они являются оптимальными решениями многокритериальных задач, так как принадлежат утопическому множеству, которое представлено на рисунке 11.

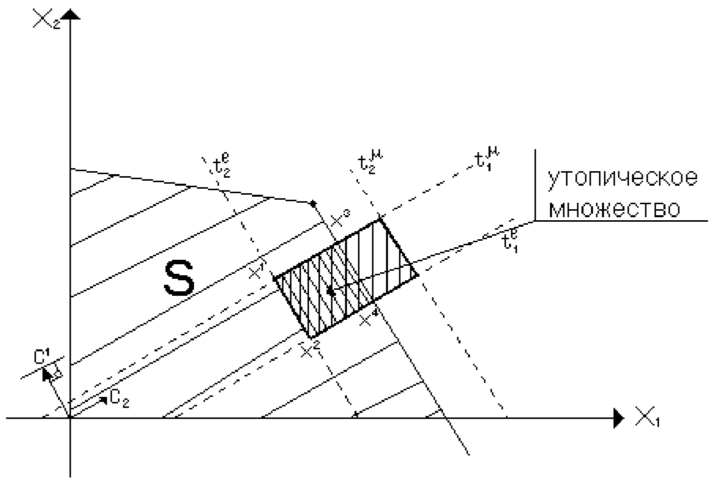


Рисунок 11. Иллюстрация к примеру

Если данная задача будет решаться обычными методами решения задач ЛП, то в качестве решения может быть сгенерировано только конечное число различных точек, и они могут представлять собой в пространстве решений следующие:

1) крайние точки S
2) точки границы области S, не являющиеся крайними, такие как а) точка z^1 из примера 1(1) б) точки x^3, x^4
3) внутренние точки S, такие как x^1, x^2.

Если же ЛПР предпочитает точку, не являющуюся крайней точкой допустимой области архимедовых задач ЦП, то её нельзя получить, не использовав специальные процедуры, связанные с изменением целевых показателей t_i.

2.3 Задачи целевого программирования с приоритетами, метод уступок

В задачах ЦП с приоритетами (лексикографическое ЦП) цели группируются по приоритетам. Цели с высшим (первым) уровнем приоритета считаются бесконечно важными по сравнению с целями со следующим (вторым) уровнем приоритета, цели со вторым уровнем приоритета – бесконечно важными по сравнению с целями с третьим уровнем приоритета и так далее.

Рассмотрим следующий пример.

Пример 5.

Цель 1: $z_1=c^1x$ - имеет первый уровень приоритета P1

Цель 2: $z_2=c^2x$ – второй уровень приоритета P2

Цель 3: $z_3=c^3x$ – третий уровень приоритета P3

Требования:

1. по первой цели вида t_1

2. по второй цели вида t_2

3. по третьей цели вида t_3

Задача ЦП с приоритетами в лексикографической форме записывается в следующем виде,

$$\text{Lex min}\ \{d_1^+, d_2^-, d_3^+ + d_3^-\}$$

при ограничениях

$$c^1x - d_1^+ \le t_1$$
$$c^2x + d_2^- \ge t_2$$
$$c^3x - d_3^+ + d_3^- = t_3$$
$$X \in S, \quad d \ge 0$$

Для решения этой задачи поиска lex min с помощью обычных задач ЦП могут потребоваться три этапа оптимизации.

На первом этапе решаем задачу:

1. $d_1^+ \to \min$

$$\begin{cases} c^1x - d_1^+ \le t_1 \\ X \in S,\ d_1^+ \ge 0 \end{cases}$$

Если в этой задаче есть альтернативные оптимумы, то формулируем и решаем задачу второго этапа:

2. $d_2^- \to \min$

$c^1x \le t_1 + (d_1^+)^*$ -оптимальное значение d_1^+, найденое на первом этапе

$c^2x + d_2^- \ge t_2$

$X \in S,\ d_2^- \ge 0$

Если в задаче второго этапа есть альтернативные оптимумы, то формулируем и решаем задачу третьего этапа:

3. $d_3^+ + d_3^- \to \min$

$c^1x \le t_1 + (d_1^+)^*$

$c^2x \ge t_2 - (d_2^-)^*$ -оптимальное значение d_2^- после второго этапа

$c^3x - d_3^+ + d_3^- = t_3$

$X \in S,\ d_3^+, d_3^- \ge 0$

где любое решение задач третьего этапа определяет лексикографический минимум в задаче ЦП с приоритетами.

Может случиться, что цели низших уровней не будут иметь шанса влиять на решение, генерируемое в задаче ЦП. Это может произойти, если на каком-либо этапе нам встретится единственное решение. Таким образом, нежелательное следствие приоритетного подхода состоит в том, что цели низших уровней могут и не иметь шанса влиять на решение, генерируемое в задаче ЦП.

Так же задачи различных этапов лексикографической оптимизации не могут решаться одновременно, так как для любого следующего этапа требуется информация об оптимальных решениях предыдущего этапа.

Данный недостаток преодолевается с помощью метода уступок, алгоритм которого аналогичен методу уступок решения задачи многокритериальной оптимизации, описанному в 1.2.

Рассмотрим задачу ЦП, область допустимых решений которой представлена на рисунке 12.

Пусть c^1 и c^2 относятся к критериям с требованием вида « ≥ » , c^1 – на первом уровне приоритета, c^2 – на втором. Следовательно решением будет являться точка x^1.

Однако, точка x^2 может быть предпочтительнее, так как небольшое уменьшение первого критерия даёт большой выигрыш в значении второго критерия.

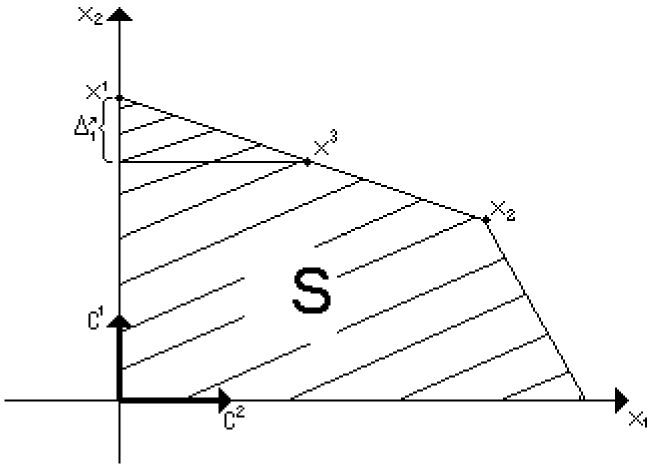


Рисунок 12. Область допустимых решений

Решая данную задачу методом уступок и выбирая уступку Δ_1, на 2-ом этапе получаем лексикографическую задачу:

$$\begin{cases} d_2^- \to \min \\ c^1x \geq t_1-(d_1^-)^*-\Delta_1 \\ c^2x+d_2^- \geq t_2 \\ x\in S,\ d_2^- \geq 0 \end{cases},$$

решением которой является точка x^3 (рисунок 12).

Успешность реализации метода уступок зависит от выбора их величин на каждом этапе проведения оптимизации.

2.5 ЦП – эффективность

Определим понятие ЦП-эффективного решения следующим образом.

Пусть $\hat{d}$ – вектор нежелательных отклонений, соответствующий точке $\hat{x} \in S$. Будем говорить, что точка $\hat{x}$ – ЦП–эффективна (в смысле минимизации отклонений), если не существует другой точки $x\in S$, которой соответствует вектор отклонений d, удовлетворяющий условиям: $d \leq \hat{d},\ d \neq \hat{d}$.

Рассмотрим пример.

Пример 6.

Цель 1: $z_1=c^1x$

Цель 2: $z_2=c^2x$

Требования:

1. по первой цели вида “$=t_1$”
2. по второй цели вида “$\geq t_1$”

$x\in S$

Геометрическая интерпретация решения представлена на рисунке 13

Для вектора нежелательных отклонений

$$d=\begin{pmatrix} d_1^+ \\ d_1^- \\ d_2^- \end{pmatrix}$$

ЦП–эффективное множество представляет собой объединение отрезков $[x^1, x^2] \cup [x^2, x^3]$. Таким образом, точка, оптимальная с точки зрения ЛПР, должна принадлежать этому множеству.

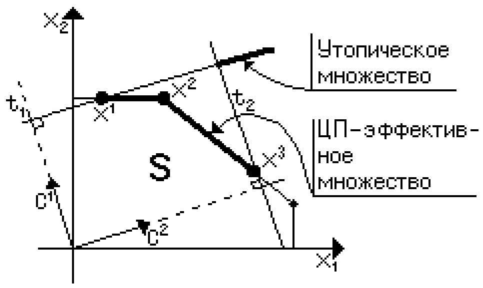


Рисунок 13. Иллюстрация к примеру

В данном примере мы снова сталкиваемся с трудностями при определении в задачах ЦП оптимальной для ЛПР точки. Используя архимедову модель, мы могли бы получить точки x^1, x^2 *или* x^3. Применив подход с использованием приоритетов, мы получили бы точки x^1 *или* x^3 в зависимости от назначенных приоритетов. Если бы точка, оптимальная для ЛПР, находилась, например, на середине отрезка (x^2, x^3), то мы не смогли бы достаточно близко подойти к ней.

Для определения ЦП – эффективного множества можно использовать векторную максимизацию, образуя минимизируемые критерии из переменных нежелательных отклонений и решая следующую задачу.

$d_1^+\to\min$,

$d_1^-\to\min$,

$d_2^-\to\min$,

при ограничениях:

$$\begin{cases} c^1x-d_1^++d_2^- =t \\ c^2x+d_2^-\geq t_2 \\ x\in S \\ d_1^+,\ d_1^-,\ d_2^-\geq 0 \end{cases}$$

2.6 Проблемы чувствительности

Обычно на практике в задачах ЦП присутствует большое число переменных отклонений. Так как они могут иметь разную размерность, то возникают большие трудности с назначением весов.

По этой причине в архимедовом ЦП мы сначала пытаемся использовать некоторые разумно назначенные штрафные веса, предполагая в дальнейшем провести анализ чувствительности с использованием других наборов весов и посмотреть, нет ли лучших решений. Часто такой подход «работает» удовлетворительно. Часто — нет. В тех случаях, когда такой подход «не работает», может, например, произойти следующее.

Пусть набор весов приводит нас к неудовлетворительному решению. Тогда некоторые веса изменяют и задачу решают снова. Иногда новое решение будет тем же, что и прежнее, так как оба набора весов соответствуют одной и той же вершине расширенного множества допустимых решений в задаче ЦП. Иногда, даже если мы чувствуем, что изменили веса разумным способом, получаем худшее решение. Иногда, несмотря на то, что мы очень слабо изменили веса, получается совершенно другое решение, потому что мы смогли «перепрыгнуть» в вершину, находящуюся с противоположной стороны большой ЦП-эффективной грани. В результате часто пользователь отказывается от архимедова подхода к решению задач ЦП, чувствуя полное крушение своих надежд по поводу возможности управлять изменением решения с помощью варьирования весов.

В задачах ЦП с приоритетами пользователи часто исследуют чувствительность путем ротации приоритетов. Если имеется r уровней приоритета, то имеется r! вариантов их ротации. Обычно пользователь выбирает небольшое число вариантов приоритета и решает задачу для каждого из них. В результате обычно получается несколько сильно отличающихся ЦП-эффективных точек из S.

Также можно провести масштабирование целевых ограничений.

В этом случае отклонения от цели измеряют в процентах.

Например: $c^i x+d_i^- \geq t_i$

Если $c^i x=20000$, $t_i=25000$, то значение $d_i^-=5000$. Таким образом, величина d_i^- составляет 20% уровня цели, следовательно, если умножим d_i на коэффициент $t_i/100$, то получим $d_i^-=20$:

$c^i x+t_i d_i^-/100 \geq t_i$

$20000+250*20 \geq 25000$.

2.7 Интерактивное целевое прграммирование

Еще одна стратегия использования целевого программирования — это применить оба подхода (архимедов и с приоритетами) в интерактивном контексте.

На первой итерации интерактивной процедуры ЛПР производит следующие действия:

1) назначает способы достижения каждой цели;

2) назначает пороговые значения для всех целей;

3) группирует цели по уровню приоритета;

4) назначает штрафные веса внутри одного уровня приоритета для тех уровней, которые определяются более чем одной целью.

Затем задача ЦП решается с использованием обычных лексикографических принципов. После того как решение получено и исследовало, к ЛПР обращаются с просьбой уточнить формулировку задачи и внести какие-нибудь подходящие изменения целей, пороговых значений, штрафных весов. Далее решается новая задача ЦП, строится решение второй итерации и т. д. Заметим, что этот подход годится и для архимедовой модели, когда имеется всего один уровень приоритета.

В пользу интерактивного подхода свидетельствует тот факт, что в случае ограниченной области S генерируются только ЦП-эффективные решения. При этом, однако, формулировка задачи может изменяться столь многими способами, что не всегда ясно, что именно нужно сделать, чтобы попытаться изменить решение в желаемом направлении. Следовательно, интерактивный метод — это весьма неконструктивный способ «зондирования» множества допустимых решений. Интерактивное ЦП существенно опирается на опыт и интуицию пользователя, используя концепции ЦП в виде некоторого средства выбора решения.

2.8 Минимизация максимального отклонения

Если какой-то уровень приоритета включает отклонения от нескольких целей, то один из возможных способов постановки задачи – минимизация максимального отклонения:

Для этого вводится минимаксная переменная $\alpha \in R$ и критерий на рассматриваемом уровне приоритета имеет вид $\alpha \to \min$.

При этом нужно выписать столько дополнительных ограничений, сколько имеется переменных отклонения на рассматриваемом уровне приоритета.

Приведем пример.

Пример 7.

Цель 1: $z_1=c^1x$ - первый уровень приоритета $z_1=t_1$;

Цель 2: $z_2=c^2x$ - второй уровень приоритета $z_2\ge t_2$;

Цель 3: $z_3= c^3x$ - второй уровень приоритета $z_3\ge t_3$;

Цель 4: $z_4=c^4x$ - второй уровень приоритета $z_4\ge t_4$;

$x\in S$.

Если мы намереваемся минимизировать максимальное отклонение на втором уровне приоритета, то лексикографическая формулировка задачи ЦП будет такой:

$$\text{Lex}\min\{(d_1^{+}+d_1^{-}),\ \alpha\}$$

При

$$c^1x-d_1^{+}+d_1^{-}=t_1$$
$$c^2x+d_2^{-}\geq t_2$$
$$c^3x+d_3^{-}\geq t_3$$
$$c^4x+d_4^{-}\geq t_4$$
$$d_2^{-}\leq\alpha$$
$$d_3^{-}\leq\alpha$$
$$d_4^{-}\leq\alpha$$
$$x\in S,\ \text{все } d\geq 0.$$

Итак, в результате проведенного анализа методов решения задач многокритериальной оптимизации можно выделить следующие проблемы многокритериальной оптимизации.

Первая проблема связана с выбором принципа оптимальности, который строго определяет свойства оптимального решения и отвечает на вопрос, в каком смысле оптимальное решение превосходит все остальные допустимые решения.

Вторая проблема связана с учетом приоритета (или различной степени важности) локальных критериев. Хотя при выборе решения и следует добиваться наивысшего качества по всем критериям, однако степень совершенства по каждому из них, как правило, имеет различную значимость.

К вышесказанному можно добавить также то, что трудности вызывает одновременное наличие в задаче многокритериального программирования качественных и количественных критериев, а именно – перевод из качественных в количественные критерии для дальнейшей оптимизации построенной математической модели. Да и сам правильный подбор весовых коэффициентов иногда сделать не так просто.

С рассмотренными выше проблемами связаны основные трудности решения задач многокритериальной оптимизации, и от того, насколько успешно они будут преодолены, во многом зависят успех и правильность выбора решения. Поэтому в процессе решения задачи непременно должно участвовать ответственное за принятие решения лицо. Следовательно, актуальным является разработка интерактивных процедур оптимизации, при которых во время процесса решения задачи производится ввод необходимой информации от ЛПР. Это может быть информация разного рода в зависимости от поставленной задачи.

3 Решение задач ЦП с использованием программного средства «Многокритериальная оптимизация: целевое программирование»

Рассмотрим решение следующей задачи с помощью программного средства «Многокритериальная оптимизация: целевое программирование».

Пример 9.

Даны две цели

$$z_1 = x_1$$
$$z_2 = x_2$$

По первой цели необходимо достичь равенства 8, по второй - равенства 7 при заданных ограничениях:

$$\begin{cases} x_1 + x_2 \le 5 \\ x_1 + 1{,}5x_2 \le 6 \\ x_1 + 2{,}5x_2 \le 8{,}75 \\ x_1 + 3{,}5x_2 \le 11{,}9 \\ x_2 \le 3{,}3 \\ 1{,}5x_1 + x_2 \le 6{,}8 \\ 2{,}5x_1 + x_2 \le 10{,}7 \\ 3{,}5x_1 + x_2 \le 14{,}65 \\ x_1, x_2 \ge 0 \end{cases}$$

Графическое отображение области допустимых значений имеет следующий вид (рисунок 1).

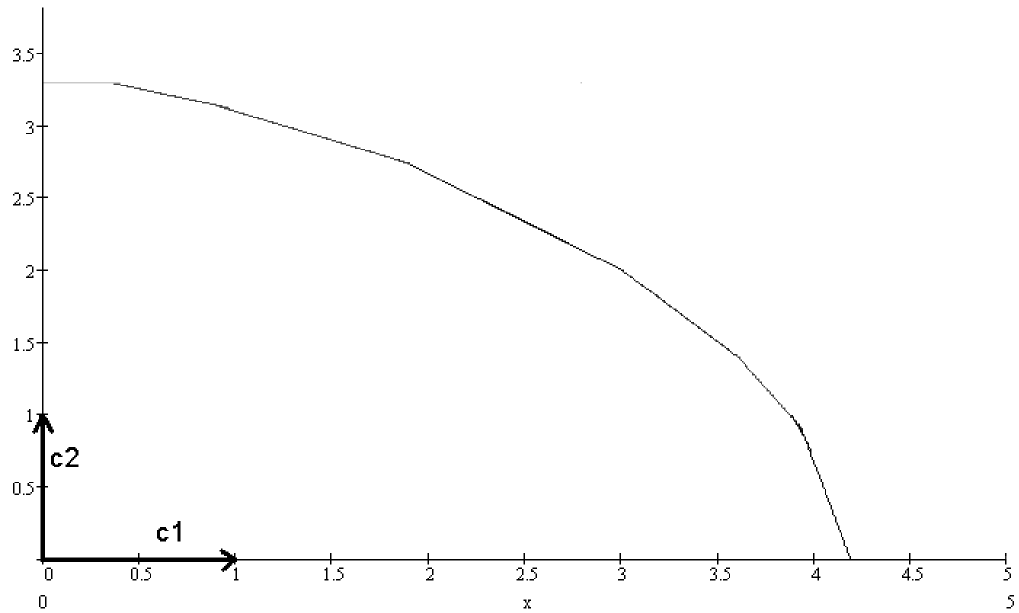


Рисунок 14 – Графическое отображение области допустимых значений

Решим данную задачу с помощью программного средства «Многокритериальная оптимизация: целевое программирование».

При запуске программного средства появится форма (рисунок 15).

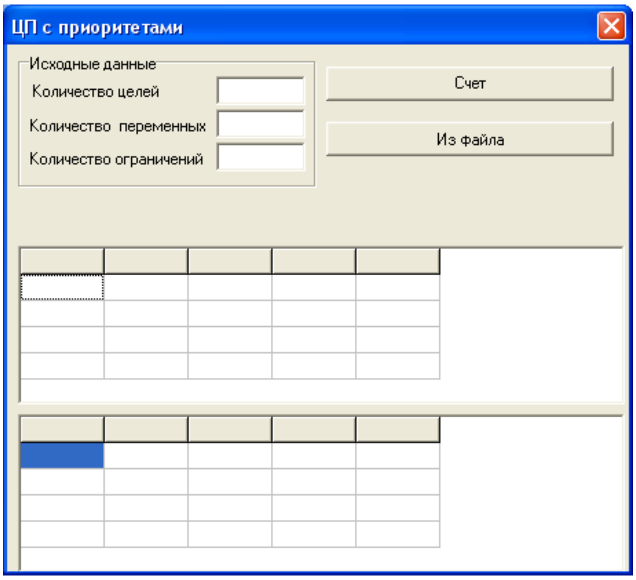


Рисунок 15 – Окно главной формы

Данные можно ввести вручную или ввести из файла. Ввод из файла организуется путем нажатия кнопки “из файла”, пользователь должен указать путь к файлу (рисунок 3). В нашем случае - файл “example1” в каталоге “cel_prog”.

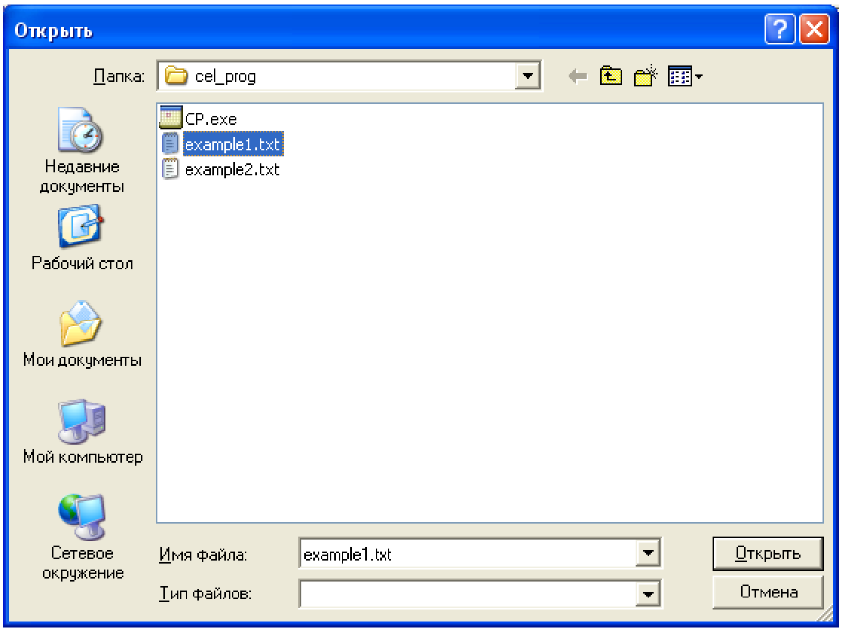


Рисунок 16 – Ввод исходных данных из файла

Введенные исходные данные представлены на рисунке 17.

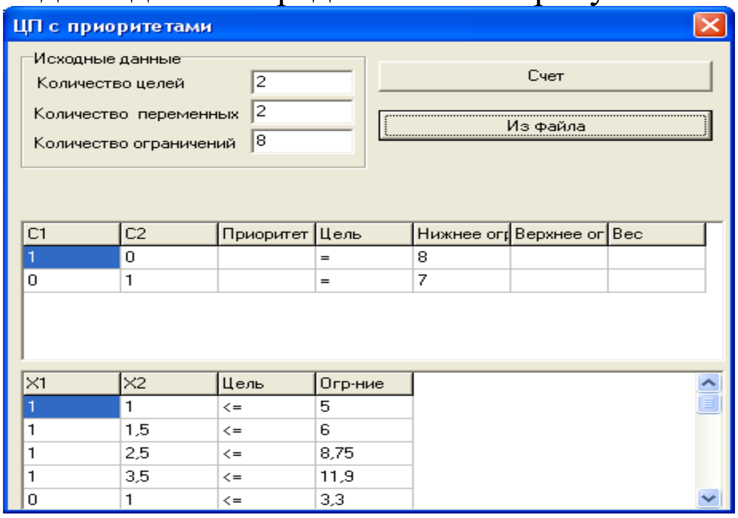


Рисунок 17 – Главная форма после ввода исходных данных

Независимо от способа ввода данных, необходимо заполнить поля столбцов «Приоритет» и «Вес». В случае, если пользователь хочет решить задачу целевого программирования без использования приоритета, в полях обоих столбцов необходимо поставить значения весов.

При задании весов для целей с одинаковыми приоритетами необходимо заполнить соответствующие поля столбца «Вес». При этом следует учитывать, что задание приоритетов целей начинается с единицы, т.е. программа не будет работать, если при задании весов единственной группе целей с одинаковыми приоритетами в соответствующих ячейках столбца «Приоритет» стоят значения, не равные единице (рисунок 18).

C1	C2	Приоритет	Цель	Нижнее огр	Верхнее ог	Вес
1	0	1	=	8		8
0	1	1	=	7		5

Рисунок 18- Ввод весов при одинаковых приоритетах

При использовании приоритетов (т.е. ввода значений в соответствующее поля столбца «Приоритет») необходимо заполнить соответствующие поля столбца «Вес», которые могут быть произвольными числами (рисунок 19).

C1	C2	Приоритет	Цель	Нижнее огр	Верхнее ог	Вес
1	0	1	=	8		1
0	1	2	=	7		1

Рисунок 19- Ввод различных приоритетов

Рассмотрим случай, при котором цели имеют одинаковый приоритет и веса, сильно разнящиеся относительно друг друга, причём приоритет первой цели больше второй (в данном случае-0,9 и 0,1 – рисунок 20).

C1	C2	Приоритет	Цель	Нижнее огр	Верхнее ог	Вес
1	0	1	=	8		0,9
0	1	1	=	7		0,1

Рисунок 20- Ввод весов 0,9 и 0,1

После ввода данных и нажатия кнопки “Счет” появится вторая форма «Результат» (рисунок 21), на которой будет представлены значения искомых переменных: $x_1 = 4{,}186, x_2 = 0,\ d_1^+ = 3{,}814, d_1^- = 0, d_2^+ = 7, d_2^- = 0$

Результат

x1	x2	d1+	d1-	d2+
4,186	0,000	3,814	0,000	7,000

Рисунок 21- Результаты вычислений при весах 0,9 и 0,1

Уменьшим разницу между весами (0,7 и 0,3 – рисунок 22).

C1	C2	Приоритет	Цель	Нижнее огр	Верхнее ог	Вес
1	0	1	=	8		0,7
0	1	1	=	7		0,3

Рисунок 22- Ввод весов 0,7 и 0,3

Программа выдаст следующие результаты (рисунок 23): $x_1 = 3{,}9, x_2 = 0{,}95,\ d_1^+ = 4{,}1, d_1^- = 0,\ d_2^+ = 6{,}05,\ d_2^- = 0$

Результат

x1	x2	d1+	d1-	d2+
3,900	0,950	4,100	0,000	6,050

Рисунок 23- Результаты вычислений при весах 0,7 и 0,3

Рассмотрим работу программы при задании одинаковых весов (рисунок 24) при равенстве приоритетов.

C1	C2	Приоритет	Цель	Нижнее огр	Верхнее ог	Вес
1	0	1	=	8		0,5
0	1	1	=	7		0,5

Рисунок 24- Ввод одинаковых весов

Получили следующие результаты (рисунок 25):

$$x_1 = 3, x_2 = 2,\ d_1^+ = 5, d_1^- = 0,\ d_2^+ = 5,\ d_2^- = 0$$

Результат

x1	x2	d1+	d1-	d2+
3,000	2,000	5,000	0,000	5,000

Рисунок 25- Результаты вычислений при одинаковых весах

Далее введём вес второй цели, ненамного превосходящий вес первой (рисунок 26).

C1	C2	Приоритет	Цель	Нижнее огр	Верхнее ог	Вес
1	0	1	=	8		0,4
0	1	1	=	7		0,6

Рисунок 26- Ввод весов 0,4 и 0,6

Результаты работы представлены на рисунке 27

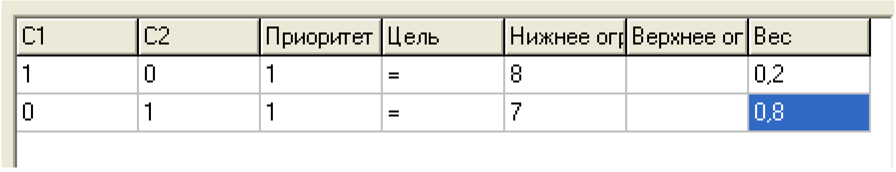
$x_1 = 1{,}875,\ x_2 = 2{,}75,\ d_1^+ = 6{,}125,\ d_1^- = 0,\ d_2^+ = 4{,}25,\ d_2^- = 0$.

Результат

x1	x2	d1+	d1-	d2+
1,875	2,750	6,125	0,000	4,250

Рисунок 27- Результаты вычислений при весах 0,4 и 0,6

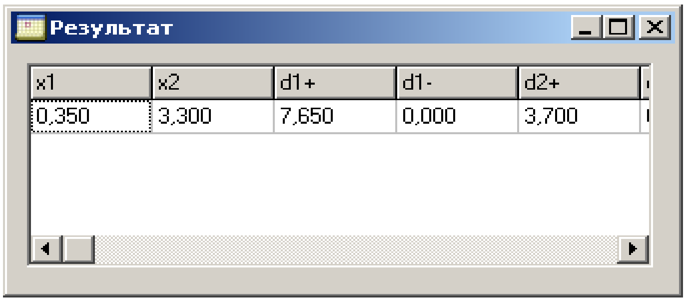
Рассмотрим случай, когда вес второй цели намного превосходит вес первой (рисунок 28).



C1	C2	Приоритет	Цель	Нижнее огр	Верхнее ог	Вес
1	0	1	=	8		0,2
0	1	1	=	7		0,8

Рисунок 28- Ввод весов 0,2 и 0,8

Программа выдаст следующий результат (рисунок 29): $x_1 = 0{,}35$, $x_2 = 3{,}3$, $d_1^+ = 7{,}65$, $d_1^- = 0$, $d_2^+ = 3{,}7$, $d_2^- = 0$



Результат

x1	x2	d1+	d1-	d2+
0,350	3,300	7,650	0,000	3,700

Рисунок 29- Результаты вычислений при весах 0,2 и 0,8

Таким образом, в результате решения задачи могут быть получены разные результаты, представляющие собой значения крайних точек области допустимых значений, что является отражением предпочтений ЛПР. Предпочтения ЛПР определяются соответствующим заданием весов и приоритетов.

Решим с помощью программного средства следующую задачу ЦП.

Пример 10.

Цель $x_1=z_1$, $z_1=8$

Цель $x_2=z_2$, $z_2=7$

При ограничениях $-x_1+2x_2\le 8$

$x_1+x_2\le 10$

$x_1,x_2\ge 0$

Рассмотрим область допустимых решений и утопическое множество в пространстве критериев. В данном случае утопическим множеством является т. (8,7), геометрическая интерпретация решения представлена на рисунке 5. Решения данной задачи – точки (4,6) и (8,2) - могут быть получены при различном задании приоритетов исходным целям.

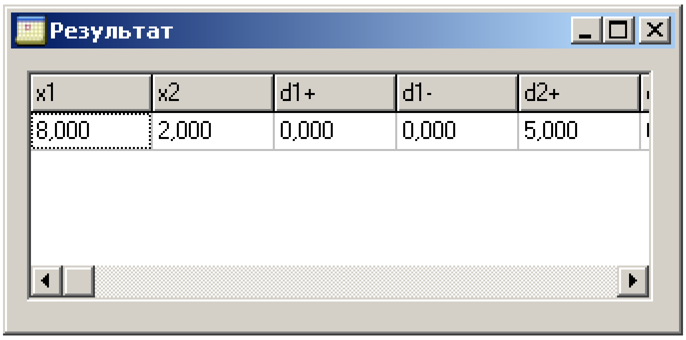
Рассмотрим работу программы для этого примера. Данные могут быть взяты из файла “example2” в каталоге “cel_prog”.

Зададим больший приоритет первой цели (рисунок 30).

C1	C2	Приоритет	Цель	Нижнее огр	Верхнее ог	Вес
1	0	1	=	8		1
0	1	2	=	7		1

Рисунок 30- Ввод приоритетов 1 и 2

Программа выдаст в качестве результата точку (8,2), значения отклонений - $d_1^+ = 0, d_1^- = 0, d_2^+ = 5, d_2^- = 0$ (рисунок 31).



x1	x2	d1+	d1-	d2+
8,000	2,000	0,000	0,000	5,000

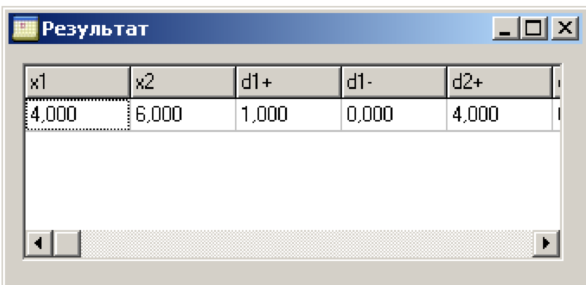
Рисунок 31 - Результаты вычислений при большем приоритете 1 цели

Рассмотрим случай большего приоритета второй цели (рисунок 32).

C1	C2	Приоритет	Цель	Нижнее огр	Верхнее ог	Вес
1	0	2	=	8		1
0	1	1	=	7		1

Рисунок 32 - Ввод приоритетов 2 и 1

Программа выдаст в качестве результата точку (4,6), значения отклонений - $d_1^+ = 1, d_1^- = 0, d_2^+ = 4, d_2^- = 0$ (рисунок 33).



x1	x2	d1+	d1-	d2+
4,000	6,000	1,000	0,000	4,000

Рисунок 33- Результаты вычислений при большем приоритете 2 цели

Пример 11.

Нефтегорск - небольшой городок, в котором проживают около 20000 человек. Городской совет разрабатывает ставки местного налогообложения. Ежегодная база налогообложения недвижимости составляет 550 млн.рублей. Ежегодная база налогообложения розничных и оптовых продаж составляет 35 и 55 млн.рублей соответственно. Ежегодное потребление городом бензина оценивается в 7.5 млн.литров. Городской совет планирует разработать систему налоговых ставок, основанную на перечисленных базах налогообложения и учитывающую следующие ограничения и требования:

1. Налоговые поступления должны составить не менее 16 млн.рублей от всех баз налогообложения;

2. Налог с розничных продаж не может превышать 10% от суммы всех собираемых налогов;

3. Налог с оптовых продаж не может превышать 20% от суммы всех налогов;

4. Налог на бензин не может превышать 2 копеек за литр.

Обозначим через $x_н, x_р, x_о$ ставки налогов (выраженные в десятичных дробях) на недвижимость, розничную и оптовую торговлю соответственно, а через $x_б$ – налог на бензин, выраженный в копейках на литр. Тогда математическая модель будет иметь вид:

$$\begin{cases} 550x_н + 35x_р + 55x_о + 0{,}075x_б \geq z_1 \quad при \; z_1 = 16 \\ 55x_н - 31{,}5x_р + 5{,}5x_о + 0{,}075x_б \geq z_2 \quad при \; z_2 = 0 \\ 110x_н + 7x_р - 44x_о + 0{,}015x_б \geq z_3 \quad при \; z_3 = 0 \end{cases}$$

$$x_б \leq 2$$

$$x_н, x_р, x_о, x_б \geq 0$$

Решим поставленную задачу с помощью программного средства.

Зададим равные приоритеты целям (рисунок 34).

C1	C2	C3	C4	Приоритет	Цель	Нижнее огр	В
550	35	55	0,075	1	>=	16	
55	-31,5	5,5	0,075	1	>=	0	
110	7	-44	0,015	1	>=	0	

Рисунок 34- Ввод данных матрицы целей

Программа выдаст в качестве результата точку (0,02; 0,046; 0,058; 0), значения отклонений - $d_1^+ = 0, d_2^+ = 0, d_3^+ = 0$ (рисунок 35).

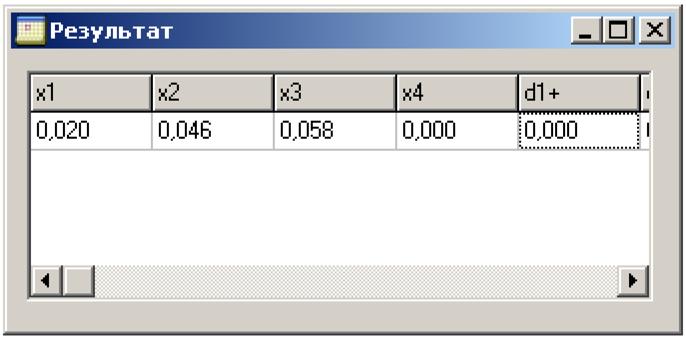


Рисунок 35 - Результаты вычислений

Таким образом, оптимальная ставка налога на недвижимость составляет 2%; ставка налога на розничную торговлю – 4,6%; ставка налога на оптовую торговлю – 5,8%; налог на бензин введен не будет.

Пример 12.

Новое рекламное агентство, в составе которого 10 рекламных агентов, получило контракт на рекламу нового продукта. Агентство может провести рекламную акцию на радио и телевидении. В таблице 1 приведены данные о количестве, охватываемых тем или иным видом рекламы, стоимость этой рекламы и количество необходимых рекламных агентов. Все эти данные приведены к одной минуте рекламного времени.

Таблица 1 – Исходные данные

	Радио	Телевидение
Рекламная аудитория (млн. человек)	4	8
Стоимость (тыс. рублей)	8	24
Количество рекламных агентов	1	2

Реклама на радио и телевидении должна охватывать не менее 45 млн. человек, но контракт запрещает использовать больше 6 минут рекламы на радио. Рекламное агентство может выделить на этот проект бюджет, не превышающий 100 тыс. рублей. Требуется определить, какое количество минут рекламного времени агентство должно купить на радио и на телевидении.

Обозначим через x_1, x_2 количество минут рекламного времени, закупленного сответственно на радио и телевидении. Таким образом, математическая модель данной задачи может выглядеть следующим образом:

$$\begin{cases} 4x_1 + 8x_2 = z_1 \quad при \; z_1 = 45 \\ 8x_1 + 24x_2 = z_2 \quad при \; z_2 = 100 \end{cases}$$

$$x_1 \le 6$$

$$x_1 + 2x_2 \le 10$$

$$x_1, x_2 \ge 0$$

Рассмотрим работу программы для этого примера.
Зададим равные приоритеты целям (рисунок 36).

C1	C2	Приоритет	Цель	Нижнее огр	Верхнее ог	Вес
4	8	1	=	45		1
8	24	1	=	100		1

Рисунок 36- Ввод данных матрицы целей

Программа выдаст в качестве результата точку (5; 2,5), значения отклонений - $d_1^+ = 5, d_1^- = 0, d_2^+ = 4, d_2^- = 0$ (рисунок 37).

Результат

x1	x2	d1+	d1-	d2+
5,000	2,500	5,000	0,000	0,000

Рисунок 37 - Результаты вычислений

Таким образом, оптимальное время рекламы на радио составляет 5 минут, на телевидении-2,5 минуты. При этом рекламная аудитория составит 5 млн. человек.

Используя разработанное программное средство «Многокритериальная оптимизация: целевое программирование» можно эффективно решать задачи целевого программирования. Руководство программисту и текст основных модулей программы представлены в приложениях А и Б соответственно.

Список использованных источников

1 **Аттетков, А.В.** Методы оптимизации: [Текст]: учебное пособие / А. В. Аттетков, С.В. Галкин, В.С. Зарубин. – М.: Изд-во МГТУ им. Н.Э, Баумана, 2001. – 440 с.

2 **Сухарев, А. Г.** Курс методов оптимизации / А.Г. Сухарев. – М.: Наука, 1986. – 325 с.

3 **Таха, Хэмди А.** Введение в исследование операций: [Текст]: пер. с англ./ Хэмди, А. Таха.- 6-е изд.- М.: Вильямс,2001.-912с.

4 **Штойер, Ральф.** Многокритериальная оптимизация. Теория вычисления и приложения / Р. Штойер. – М.: Радио и связь, 1992. – 504 с.

Приложение А

Руководство программисту
Описание основных процедур

Рассмотрим все процедуры и функции, используемые в програме.

Модуль MainUnit:

В данном модуле реализуются следующие процедуры.

1) procedure simplex(napr,n,m:word; znak:Tmass; koef,ogran:Tvect; mo:Tmatr; var resh:Tvect; var rez:real);

Входные данные:

napr – направление оптимизации (0-минимизация; 1- максимизация);

n – количество переменных;

m – количество ограничений;

znak – вектор знаков при ограничениях(1-≤ ; 2-≥ ;3-=);

koef – коэффициенты целевой функции;

ogran – вектор свободных членов ограничений;

mo – матрица коэффициентов ограничений.

Выходные данные

resh – вектор решения, полученный с помощью симплекес-метода;

rez – значение целевой функции в оптимальной точке.

2) procedure TForm1.Button1Click(Sender: TObject);

Процедура, формирующая данные для последующего решения задачи ЦП с приоритетами (учитывая вес цели) с помощью симплекс-метода: матрицу ограничений, целевую функцию.

3) procedure TForm1.EtarChange(Sender: TObject);

Процедура для формирования таблицы (количество строк, равное количеству целей) ввода данных целей на главной форме.

4) procedure TForm1.EnChange(Sender: TObject);

Процедура для формирования таблицы (количество столбцов, равное количеству переменных) ввода данных целей и ограничений на главной форме.

5) procedure TForm1.EogrChange(Sender: TObject);

Процедура для формирования таблицы (количество строк, равное количеству ограничений) ввода данных ограничений на главной форме.

6) procedure TForm1.sgDrawCell(Sender: TObject; ACol, ARow: Integer;
Rect: TRect; State: TGridDrawState);

Процедура, реализующая отображение таблицы ввода данных целей на главной форме.

7) procedure TForm1.SGXDrawCell(Sender: TObject; ACol, ARow: Integer;
Rect: TRect; State: TGridDrawState);

Процедура, реализующая отображение таблицы ввода данных ограничений на главной форме.

8) procedure TForm1.cbZnakChange(Sender: TObject);

Процедура, реализующая отображение выбора знака цели.

9) procedure TForm1.cbZnakxChange(Sender: TObject);

Процедура, реализующая отображение выбора знака ограничений.

10) procedure TForm1.BitBtn1Click(Sender: TObject);

Процедура, реализующая ввод данных из файла.

Модуль Unit2:

Модуль для отображения формы с результатами работы программы - оптимальными значениями исходных переменных.

Модуль NewSimplex

Модуль, реализующий работу симплекс-метода.

В данном модуле реализуются следующие процедуры:

1) procedure fiktiv(n,m:word; znak:Tmass; var mo:Tmatr; var nf:word);

Процедура приведения базиса к каноническому виду.

2) procedure iskystv(nf,m:word; znak:Tmass; var mo:Tmatr; var ni:word);

Процедура формирования искусственного базиса.

3) procedure nachalbazis(n,nf,m:word; znak:Tmass; var baz:Tmass);

Процедура формирования начального базиса.

4) procedure resheniematr(m,nn:word; var mo:Tmatr; var c1:Tvect; var baz:Tmass; var q:boolean);

Процедура, формирующая матричное решение симплекс-метода.

Модуль Typess

Модуль, содержащий описания всех типов данных, используемых в программе.

Приложение Б

Код программы

```
unit Unit1;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, Grids, StdCtrls, NewSimplex,Typess;
type
   mas=array of real;
   matr=array of mas;
  TForm1 = class(TForm)
    Button1: TButton;
    Label1: TLabel;
    kcel: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    kper: TEdit;
    kogr: TEdit;
    Memo1: TMemo;
    Tabl: TStringGrid;
    Button2: TButton;
    Button3: TButton;
    w: TStringGrid;
    Label4: TLabel;
    Memo2: TMemo;
    Label5: TLabel;
    Button4: TButton;
    Pr: TStringGrid;
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
   n,m,k:word;
   znak:Tmass;
```

```
  koef,ogran:Tvect;
  mo:Tmatr;
  resh:Tvect;
  rez:real;
implementation

{$R *.dfm}

procedure TForm1.Button2Click(Sender: TObject);
 var i:integer;
 begin
   tabl.Visible:=true;
   m:=strtoint(kcel.Text)+strtoint(kogr.Text);
   Tabl.RowCount:=m+1;
   n:=strtoint(kper.Text);
     Tabl.ColCount:=n+2;
     for i:= 0 to n-1 do
       Tabl.Cells[i,0]:='X'+inttostr(i+1);
       Tabl.Cells[n,0]:='Знаки';
       Tabl.Cells[n+1,0]:='Ограничение';
  end;

procedure TForm1.Button3Click(Sender: TObject);
 var i,s:integer;
 begin
 for i:=0 to strtoint(kcel.Text)-1 do
 s:=strtoint(pr.cells[i,0])+s;
 if s=strtoint(kcel.Text)
   then
   begin
     w.Visible:=true;
     w.ColCount:=0;
     label4.Visible:=true;
     for i:=1 to strtoint(kcel.text) do
     if (tabl.Cells[n,i]='1') or  (tabl.Cells[n,i]='2') then
     w.ColCount:=w.ColCount+1
     else w.ColCount:=w.ColCount+2;
     w.ColCount:=w.ColCount-1;
   end
   else;
 end;
procedure TForm1.Button1Click(Sender: TObject);
 var i,j,q,v,s,f,b:integer;
 t,l:array[1..20,1..20] of real;
 zn,zam:array[1..20] of integer;
```

```
begin
  n:=strtoint(kper.Text);
  m:=strtoint(kcel.text)+strtoint(kogr.Text);
  k:=strtoint(kcel.Text);
  for i:=1 to m   do
    begin
      for j:=0 to n+1 do
      t[i,j+1]:=strtoint(tabl.Cells[j,i]);
    end;
       for i:=1 to m do
       zn[i]:=strtoint(tabl.Cells[n,i]);

  if w.Visible=false
  then
    begin
          q:=0;
          k:=1;
          for b:=1 to strtoint(kcel.Text) do
      begin
              for i:=1 to 20 do
              resh[i]:=0;
              if (zn[1]=1) or (zn[1]=2) then
              w.ColCount:=1
              else w.ColCount:=2;
      {}
        for i:=1 to m do
          begin
              znak[i]:=zn[i];
              ogran[i]:=t[i,n+2];
          end;

        for i:=1 to n do
            koef[i]:=0;
        for i:=n+1 to n+w.ColCount do
          { koef[i]:=strtofloat(w.Cells[i-n-1,0]);  }
            koef[i]:=1;

          for j:=1 to n do
          mo[1,j]:=t[1,j];

          for i:=k+1-q to m do
          begin
          for j:=1 to n do
          mo[i,j]:=t[i,j];
          end;
```

```
       for i:=1 to m do
        begin
            for j:=n+1 to n+w.ColCount do
            mo[i,j]:=0;
        end;
    s:=n+w.colcount;
    f:=n;
   if zn[i]=1
    then
        begin
          f:=f+1;
          mo[i,f]:=1;
        end
    else
      if zn[i]=2 then
        begin
             f:=f+1;
             mo[i,f]:=-1;
        end
    else
      if zn[i]=3 then
        begin
            f:=f+1;
            mo[i,f]:=1;
            f:=f+1;
            mo[i,f]:=-1;
        end
else;
        v:=m-k+q+1;

simplex(0,s,v,znak,koef,ogran,mo,resh,rez);
q:=q+1;
if zn[1]=1 then
begin
  t[1,n+2]:=t[1,n+2]+resh[n+1];
end
  else if zn[1]=2 then
  t[1,n+2]:=t[1,n+2]-resh[n+1]
  else   t[1,n+2]:=t[1,n+2]-resh[n+1]+resh[n+2];

for i:=1 to m-1  do
begin
  for j:=1 to n+2 do
  l[i,j]:=t[i+1,j];
end;
```

```
 for j:=1 to n+2 do
   l[m,j]:=t[1,j];

  for i:=1 to m  do
 begin
   for j:=1 to n+2 do
   t[i,j]:=l[i,j];
 end;

 for i:=1 to m-1 do zam[i]:=zn[i+1];
 zam[m]:=zn[1];
 for i:=1 to m do zn[i]:=zam[i];
 end;
end
  else
begin
 n:=strtoint(kper.Text);
 m:=strtoint(kcel.text)+strtoint(kogr.Text);
 for i:=1 to m do
 begin
   znak[i]:=strtoint(tabl.Cells[n,i]);
   ogran[i]:=strtoint(tabl.Cells[n+1,i]);
 end;

 for i:=1 to n do
 koef[i]:=0;
 for i:=n+1 to n+w.ColCount do
 koef[i]:=strtofloat(w.Cells[i-n-1,0]);
 {koef[i]:=1;}
 for i:=1 to strtoint(kcel.text) do
 begin
     for j:=0 to n-1 do
     mo[i,j+1]:=strtofloat(tabl.Cells[j,i]);
 end;

 for i:=strtoint(kcel.Text)+1 to m do
 begin
     for j:=0 to n-1 do
     mo[i,j+1]:=strtofloat(tabl.Cells[j,i]);
 end;

 for i:=1 to m do
   begin
   for j:=n+1 to n+w.ColCount do
```

```
        mo[i,j]:=0;
      end;
    s:=n+w.colcount;
    f:=n;
    for i:=1 to strtoint(kcel.text) do
    if tabl.Cells[n,i]='1' then
      begin
        f:=f+1;
        mo[i,f]:=1;
      end
    else
    if tabl.Cells[n,i]='2' then
      begin
        f:=f+1;
        mo[i,f]:=-1;
      end
    else
    if tabl.Cells[n,i]='3' then
    begin
    f:=f+1;
    mo[i,f]:=1;
    f:=f+1;
    mo[i,f]:=-1;
    end
    else;
    simplex(0,s,m,znak,koef,ogran,mo,resh,rez);
    end;
    memo2.Visible:=true;
    for i:=1 to n do memo2.Lines.Add('x'+inttostr(i)+'='+floattostr(resh[i]));
    memo2.Lines.Add('Значение целевой функции='+floattostr(rez));
       {}
   for i:=1 to 20 do resh[i]:=0;
  end;
procedure TForm1.Button4Click(Sender: TObject);
begin
  Pr.Visible:=true;
  Label5.Visible:=true;
  Pr.ColCount:=strtoint(kcel.Text);
  W.Visible:=false;
  W.ColCount:=1;
  label4.Visible:=false;
end;
end.
```