1. Задача математического программирования.

Задача математического программирования — это математическая модель поиска наилучшего (оптимального) решения среди множества возможных.

Основные компоненты задачи:

1. Управляемые переменные ($x_1, x_2, \ldots, x_n$) — это параметры, которые могут изменяться (количество продуктов в рационе, товаров).
2. Целевая функция ($f(X)$) — это критерий эффективности. Задача может стоять как минимизировать её (затраты, время, риск), так и максимизировать (прибыль, КПД, надежность).
3. Ограничения — условия, которым должны удовлетворять переменные (бюджет, физическая неотрицательность веса).

В общем виде задача записывается следующим образом. Найти вектор $X=(x_1, x_2, \ldots, x_n)$, такой что $f(x) \to min(или\, max)$, при условиях (ограничениях):

- $g_i(X) \leq 0, i=1,\ldots,m$ (неравенства)
- $h_j(X)=0, j=1,\ldots,k$ (равенства)
- $X \in D$ (область допустимых значений, например $x_i \geq 0$)

Множество всех точек X, которые удовлетворяют ограничениям, называется допустимым множеством (или областью).

2. Классификация задач математического программирования.

1. Если целевая функция является скалярной (состоит из одной функции), то задача называется задачей однокритериальной оптимизации.
2. Если целевая функция состоит из двух и более функций, то задача называется задачей многокритериальной оптимизации.
3. Задача дискретной (комбинаторной) оптимизации: D (множество допустимых решений) — объединение изолированных точек или пересекающихся множеств. Пример: задача коммивояжера (есть несколько городов и расстояния между ними, коммивояжер должен посетить каждый город ровно один раз и вернутся в изначальный город за минимальное расстояние)
4. Задача непрерывной оптимизации (в противном случае дискретной задаче). Пример: транспортная задача.
5. Задача линейного программирования — целевая функция и функции, определеяющие множество D, линейные.
6. Задача выпуклого программирования — указанные функции выпуклы для задачи минимизации, т.е. в задаче любой локальный минимум является глобальным.

3. Классификация методов решения задач математического программирования.

По типу математической модели:

1. Методы ЛП: если функции линейны. Пример: симплекс-метод.
2. Методы НЛП (нелинейного программирования): если есть кривые линии или поверхности.

3. Методы дискретного (целочисленного) программирования: когда переменные — целые числа. Пример: метод ветвей и границ, метод отсечений (Гомори).
4. Методы динамического программирования: когда задача разбивается на этапы по времени (принцип оптимальности Беллмана).

По наличию ограничений:

1. Методы безусловной оптимизации: поиск минимума во всем пространстве R^n.
2. Методы условной оптимизации:
   1. Методы штрафных и барьерных функций: превращают задачу с ограничениями в задачу без ограничений, добавляя штраф за выход за границы.
   2. Метод множителей Лагранжа.
   3. Методы проекции градиента: когда на каждом шаге мы «спроектированы» обратно в допустимую область.

По характеру поиска:

1. Детерминированные методы: всегда выдают один и тот же результат при одинаковых входных данных. Например: симплекс-метод.
2. Стохастические (случайные) методы: используют элемент случайности. Например: эвристические методы, генетические алгоритмы.

4. Задача оптимального планирования производства.

Предприятие планирует выпускать n типов продукции из имеющихся m видов ресурсов. Известны затраты a_{ij} $i-го$ ресурса на выпуск единицы объема продукции $j-го$ типа, запас $i-го$ ресурса b_i и доход c_j от реализации единицы объёма продукции $j-го$ типа, $i=1,\dots,m$; $j=1,\dots,n$. Требуется найти план производства продукции, при котором суммарный доход будет максимальным.

Построим математическую модель задачи. Обозначим через x_j объём продукции $j-го$ типа, $j=1,\dots,n$. Тогда план выпуска продукции можно записать в виде вектора $x=(x_1,x_2,\dots,x_n)$, причем все x_j должны быть неотрицательны. Работая по этому плану, предприятие не может израсходовать больше имеющихся у него ресурсов. Заметим, что ресурсы могут быть истрачены не полностью. Таким образом, план x должен удовлетворять ограничениям: $a_{i1}x_1+a_{i2}x_2+\dots+a_{i¿}x_n\le b_i$; $i=1,\dots,m$. Суммарный доход от реализации всей продукции, произведенной в соответствии с планом x, равен: $f(x)=c_1x_1+c_2x_2+\dots+c_nx_n$.

Таким образом получаем задачу линейного программирования.

5. Задача линейного программирования.

Задачи линейного программирования являются математическими модели многих практических задач, возникающих в экономике, планировании, управлении и других областях. Кроме того, в ряде случаев решения задачи линейного программирования является этапом решения более сложной задачи оптимизации (например, задачи линейного программирования).

Общая задача линейного программирования имеет вид:

$f(x)=\sum_{(j=1)}^{n} c_j x_j \rightarrow max(min)$

$\sum_{(j=1)}^{n} a_{ij} x_j = b_i,\ i=1,\dots,m,$

$x_j \geq 0$ для некоторых $j \in 1,\dots,n,$

где под знаком = подразумеваются один из знаков <=, =, >=, причем в одной задаче ЛП могут встречаться знаки всех трех типов.

Функция f называется целевой, а условия — ограничениями задачи. Вектор $x=(x_1, x_2, \dots, x_n)$, удовлетворяющий ограничениям задачи, называется допустимым решением, или планом задачи ЛП. Множество всех допустимых решений будем обозначать через D. Геометрически множество D представляет собой выпуклое многогранное множество.

Вектор $x* \in D$ называется оптимальны решением (оптимальным планом) задачи ЛП, если $f(x*¿) \geq f(x)$ для любого $x \in D$ в задаче максимизации или $f(x*¿) \leq f(x)$ для любого $x \in D$ в задаче минимизации. Данная задача называется разрешимой, если имеет оптимальное решение. В противном случае задача ЛП неразрешима. Неразрешимость задачи ЛП может быть связана либо с противоречивостью ограничений (множество допустимых решений D пусто), либо с неограниченностью целевой функции на множестве D.

6. Графический метод решения задачи ЛП.

Суть метода заключается в геометрической интерпретации задачи:

1. Ограничения представляют собой прямые линии на плоскости. Каждое ограничение-неравенство отсекает лишнюю часть плоскости, оставляя допустимую полуплоскость.
2. Пересечение всех этих полуплоскостей образует область допустимых решения (ОРД). Это всегда выпуклый многоугольник (или бесконечная область).
3. Целевая функция ($Z=c_1 x_1 + c_2 x_2$) на графике выглядит как прямая линия. Мы перемещаем эту линию параллельно самой себе, пока она не коснется крайней точки нашего многоугольника.

Когда его можно применять? Когда в задаче 2 переменные — мы работаем на обычной плоскости. Сложнее, когда 3 переменные, т.к. надо строить объемные фигуры.

Алгоритм решения:

1. Рисуем систему координат x_1 и x_2. При условии $x_1, x_2 \geq 0$ мы работаем только в первой четверти.
2. Каждое ограничение вида $a_{i1} x_1 + a_{i2} x_2 \leq b_i$ превращаем в уравнение прямой ($=b_i$) и строим её по двум точкам.

3. Для каждого неравенства определяем нужную сторону (полуплоскость). Пересечение всех сторон даст нам многоугольник решений.
4. Строим вектор $c=(c_1, c_2)$, координаты которого — коэффициенты при x в целевой функции. Этот вектор показывает направление наискорейшего роста функции.
5. Проводим прямую, перпендикулярную вектору c. Это наша целевая функция.
6. Поиск оптимума:
    1. Для максимизации: двигаем линию в направлении вектора c до последний точки касания с ОДР.
    2. Для минимизации: двигаем линию против направления вектора c до первой точки касания.
7. Координаты найденной точки (вершины) и будут оптимальным решением. Если точка — это пересечение двух прямых, решаем систему из этих двух уравнений.

7. Каноническая задача линейного программирования.

Канонической задачей ЛП (КЗЛП) называется задача вида:

$f(x)=\sum_{(j=1)}^{n} c_j x_j \rightarrow max$

$\sum_{(j=1)}^{n} a_{ij} x_j = b_i,\ i=1,\dots,m,$

$x_j \ge 0,\ j=1,\dots,n.$

Две задачи ЛП являются эквивалентными, если каждому допустимому решению одной задачи соответствует некоторое допустимое решение другой задачи, и наоборот; причем такое же условие выполняется для оптимальных решений.

Для любой задачи ЛП легко построить эквивалентную ей КЗЛП, по оптимальному решению которой можно восстановить оптимальное решение исходной задачи. Рассмотрим правила построения КЗЛП на примере.

Дана задача ЛП:

$f(x)=x_1-x_2 \rightarrow min$

$3x_1+x_2 \le -2$

$x_1-2x_2 \ge 1$

$-x_1+x_2=-1$

$x_1 \ge 0$

Для получения КЗЛП надо перейти от минимума в целевой функции к максимуму, от ограничений-неравенств — к ограничениям-равенствам и от переменных без ограничения на знак — к неотрицательным переменным. Переход от минимума к максимуму (и наоборот) достигается умножением целевой функции на -1. Первое (и второе) ограничение можно превратить в равенство, добавив (соответственно отняв) к его левой части новую неотрицательную переменную x_3 (x_4). Эти переменные называются дополнительными. Заметив, что в рассматриваемой задаче нет ограничений на знак переменной x_2, т.е. она может принимать как положительные, так и отрицательные значения. Такую переменную можно представить в виде разности двух неотрицательных переменных: $x_2 = x'_2 - x''_2$, где $x'_2, x''_2 \geq 0$. После выполнения этих преобразований задача примет вид:

$$f(x) = -x_1 + x'_2 - x''_2 \rightarrow max$$
$$3x_1 + x'_2 - x''_2 + x_3 = -2$$
$$x_1 - 2x'_2 + 2x''_2 - x_4 = 1$$
$$-x_1 + x'_2 - x''_2 = -1$$
$$x_1, x'_2, x''_2, x_3, x_4 \geq 0$$

Эта задача является канонической. Аналогичным способом любая задача ЛП может быть приведена к канонической форме. При этом полученная КЗЛП будет эквивалентна исходной.

8. Базисные и опорные решения.

Система линейных уравнений называется системой с базисом, если в каждом уравнении имеется переменная, которая входит в него с коэффициентом $+1$ и отсутствует в остальных уравнениях. Такие переменные называются базисными и образуют базис, а остальные — небазисными. Геометрически каждому базису соответствует вершина множества D.

Приведенной задачей ЛП (ПЗЛП) называется такая каноническая задача ЛП, в которой:

1. Систему уравнений есть система с базисом.
2. Целевая функция $f(x)$ выражена только через небазисные переменные.
3. Все правые части уравнений неотрицательны: $b_i \geq 0$, $i = 1, \ldots, m$.

Отметим, что не для всякой задачи ЛП существует эквивалентная ей ПЗЛП. И даже если она существует, то получить её труднее, чем каноническую задачу ЛП. Однако во многих случаях (например, в задаче оптимального планирования производства) построенная для исходной задачи КЗЛП оказывается также и приведенной.

Базисное решение — это вектор X, полученный при условии, что все небазисные переменные равны 0.

Опорное решение — это такое базисное решение, в котором значения всех переменных неотрицательны ($x_i \geq 0$).

9. Специальная задача линейного программирования.

Это такая каноническая задача ЛП, в которой:

1. Система уравнений есть система с базисом.
2. Целевая функция выражена через небазисные переменные.
3. Правые части системы линейных ограничений неотрицательны.

Не для каждой задачи ЛП существует эквивалентная ей специальная задача ЛП.

10. Симплекс метод.

1. Привести задачу ЛП к виду приведенной задачи ЛП.
2. Составить начальную симплекс таблицу: (коэффициенты c_n записываются с противоположным знаком)

		x_1	$\cdots$	x_q	$\cdots$	x_n	x_{n+1}	x_{n+2}	$\cdots$	x_{n+m}
f	c_0	c_1	$\cdots$	c_q	$\cdots$	c_n	0	0	$\cdots$	0
x_{n+1}	b_1	a_{11}	$\cdots$	a_{1q}	$\cdots$	a_{1n}	1	0	$\cdots$	0
x_{n+2}	b_2	a_{21}	$\cdots$	a_{2q}	$\cdots$	a_{2n}	0	1	$\cdots$	0
$\cdots$	.	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
x_{n+p}	b_p	a_{p1}	$\cdots$	a_{pq}	$\cdots$	a_{pn}	0	0	$\cdots$	0
$\cdots$	.	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
x_{n+m}	b_m	a_{m1}	$\cdots$	a_{mq}	$\cdots$	a_{mn}	0	0	$\cdots$	1

3. Проверить на оптимальность ($c_j \geq 0$ для любого $j=1,\ldots,n+m$). Если оптимально, то решение найдено.
4. Если существует столбец такой, что $c_i<0$ и $a_{iq} \leq 0$ для любого $i=1,\ldots,m$, то целевая функция неограничена сверху на множестве допустимых решений, следовательно задача неразрешима.
5. Столбец с номером q выбирается в качестве ведущего, если $c_q<0$ (если несколько таких, то любой).
6. Строка с номером p выбирается в качестве ведущей, если b_i/a_{iq} — минимальное для всех $i=1,\ldots,m$ при выбранном q, а $a_{iq}>0$.
7. Элемент a_{pq} является ведущим элементом. Переменная x_q становится базисной (замещает строку p в таблице), а переменная с строки p — небазисной (уход из базиса).
8. Новый элементы ведущей строки вычисляются делением на ведущий элемент. Новые элементы остальной таблицы высчитываются по правилу прямоугольника. Новый элемент = старый элемент — (произведение соседних угловых элементов) / ведущий элемент.
9. Обратно на шаг 3.

Пример начальной симплекс таблицы:

$$
\begin{aligned}
&f(x) = 3x_1 + 3x_2 \to max \\
&\qquad 3x_1 + 2x_2 + x_3 = 10, \\
&\qquad 2x_1 + 4x_2 + x_4 = 12, \\
&\qquad 6x_1 + x_5 = 16, \\
&\qquad x_1, x_2, x_3, x_4, x_5 \geq 0.
\end{aligned}
$$

			$\downarrow$			
		x_1	x_2	x_3	x_4	x_5
f	0	-3	-3	0	0	0
x_3	10	3	2	1	0	0
$\to$ x_4	12	2	4^*	0	1	0
x_5	16	6	0	0	0	1

11. Метод искусственного базиса.

Если задача не является приведенной, то можно сделать так.

Сначала в каждой ограничение добавляется переменная t_m, а целевая функция (новая h) выражается через эти t_m. Затем из ограничений выражаются t_m и подставляются в функцию h (а именно $h(t)=-(t_1+t_2+\ldots+t_n)$) — получается вспомогательная задача ЛП.

Теорема. Каноническая задача ЛП имеет допустимое решение тогда и только тогда, когда оптимальное значение целевой функции $h*(x,t)$ вспомогательной задачи равно нулю.

Следствие. Если множество допустимых решений задачи непусто, то существует эквивалентная ей приведенная задача ЛП, которая может быть найдена из последней симплексной таблицы вспомогательной задачи.

1. Построить и решить вспомогательную задачу ЛП.
2. Если $h*¿<0$, то исходная задача ЛП не имеет допустимых решений и алгоритм заканчивает работу.
3. Если среди базисных переменных в последней симплексной таблице вспомогательной задачи имеется искусственная переменная t_i, то с помощью гауссовых преобразований над строками заменить её переменной x_j для некоторого j. После этого вычеркнуть из таблицы столбец, соответствующий переменной t_i, и строку, соответствующую целевой функции $h(x,t)$. Строки полученной таблицы и целевая функция $f(x)$ определяют приведенную задачу ЛП, эквивалентную исходной.

$f(x)$ из $h(x,t)$ получается следующим образом. Из конечной таблицы для вспомогательной задачи берутся базисные переменные (строки все, кроме самой функции) и записываются с свободным членом и коэффициентами помноженными на указанные x (без t, их не учитываем вообще). Затем эти выраженные из таблицы x подставляются в исходную функцию $f(x)$ и уже с данной функцией и ограничениями из последней таблицы для вспомогательной задачи (без столбцов t) решается задача.

12. Двойственная задача линейного программирования.

С каждой задачей ЛП тесно связана другая задача ЛП, которая называется двойственной.

Построение двойственной задачи из произвольной задачи ЛП. Предварительно необходимо согласовать целевую функцию и знаки ограничений: если целевая функция на максимум, то знаки ограничений должны быть <= или =; если на минимум, то >= или =.

1. Если прямая задача на максимум, то двойственная ей — на минимум (и наоборот).

2. Каждому линейному неравенству прямой задачи соответствует неотрицательная переменная двойственной задачи; уравнениям соответствуют переменные без ограничения в знаке (и наоборот).
3. Правые части системы ограничений прямой задачи являются коэффициентами целевой функции двойственной задачи (и наоборот).
4. Коэффициенты системы ограничений при переменной x_j прямой задачи являются коэффициентами ограничения двойственной задачи, соответствующего это переменной (и наоборот).

Пример прямой и двойственной задачи:

$$
\begin{array}{ll}
\qquad\qquad (\mathrm{I}) & \qquad\qquad (\mathrm{II}) \\
f(x) = 3x_1 + 3x_2 \to max & g(y) = 10y_1 + 12y_2 + 16y_3 \to min \\
3x_1 + 2x_2 \le 10, & \quad y_1 \ge 0, \\
2x_1 + 4x_2 \le 12, & \quad y_2 \ge 0, \\
6x_1 \qquad\quad \le 16, & \quad y_3 \ge 0, \\
\ x_1 \ge 0, & 3y_1 + 2y_2 + 6y_3 \ge 3, \\
\ x_2 \ge 0, & 2y_1 + 4y_2 \qquad\quad \ge 3.
\end{array}
$$

13. Теоремы двойственности.

Теорема 1. (Основное неравенство двойственности) Если задачи I и II разрешимы, то для любых допустимых решений x и y этих задач соответственно справедливо неравенство $f(x) \le g(y)$.

Теорема 2. (Первая теорема двойственности) Если одна из пары двойственных задач I, II разрешима, то разрешима и другая задача, причем оптимальные значения целевых функций совпадают, т.е. $f(x*¿)=g(y*¿)$, где $x*¿$, $y*¿$ - оптимальные решения задач I и II соответственно.

Следствие 1. Допустимое решение прямой задачи $x*¿$ является оптимальным тогда и только тогда, когда существует допустимое решение двойственной задачи $y*¿$, при котором $f(x*¿)=g(y*¿)$.

Говорят, что вектора x, y удовлетворяют условиям дополняющей нежесткости, если при подстановке этих векторов в любую пару сопряженных неравенств в хотя бы одно из них обращается в равенство. Это означает, что следующие произведения, которые называются характеристическими, равны нулю:

$$\left(\sum_{(j=1)}^{n} a_{ij} x_j - b_i\right) y_i = 0,\ i = 1, \dots, m,$$

$$x_j \left(\sum_{(i=1)}^{m} a_{ij} y_i - c_j\right) = 0,\ j = 1, \dots, n.$$

Теорема 3. (Вторая теорема двойственности) Допустимые решения $x*¿$, $y*¿$ являются оптимальными для задач I. II соответственно тогда и только тогда, когда они удовлетворяют УДН.

Следствие 2. (Второй критерий оптимальности) Допустимое решение прямой задачи $x*¿$ является оптимальным тогда и только тогда, когда существует допустимое решение двойственной задачи $y*¿$, такое, при котором пара ($x*¿$, $y*¿$) удовлетворяет УДН.

Теорема 4. Если множества допустимых решений прямой и двойственной задач не пусты ($D_I != !0, D_{II} != !0$), то обе задачи разрешимы.

Теорема 5. (Критерий неограниченности целевой функции) Целевая функция прямой задачи не ограничена сверху на непустом множестве допустимых решений тогда и только тогда, когда множество допустимых решений двойственной задачи пусто.

14. Экономическая интерпретация двойственной задачи

Рассмотрим задачу оптимального планирования производства и двойственную к ней задачу ЛП. Предположим, что при изучении вопроса об использовании ресурсов появилась возможность их продажи. Спрашивается, какую минимальную цену нужно установить за единицу объема каждого сырья, чтобы доход от продажи всего сырья был не меньше доходов от реализации продукции, которая может быть выпущена из этого сырья.

Значения переменных y_i можно интерпретировать как некоторые цены ресурсов. Их оптимальные значения называются объективно обусловленными оценками ресурсов.

Целевая функция выражает суммарную оценку всех имеющихся запасов ресурсов. Согласно первой Т двойственности для оптимальных векторов $x*¿$, $y*¿$ имеет место равенство $f(x*¿)=g(y*¿)$. Это означает, что максимальный доход от продажи всей готовой продукции совпадает с минимальной суммарной оценкой всех ресурсов.

15. Дефицитные и недефицитные ресурсы.

На примере конкретном. Значения дополнительных переменных x_3, x_4, x_5 можно интерпретировать как количества неизрасходованного сырья соответствующего вида. Поскольку в последней симплексной таблице $x_3=0$, $x_4=0$, то первый и второй ресурсы при оптимальном плане расходуются полностью. Такие ресурсы называются дефицитными, и их оценки $y*¿_1=3/4$, $y*¿_2=3/8$ — положительны (но иногда они могут быть равны 0).

Оптимальное значение переменной x_5 не равно нулю ($x*¿_5=4$), т.е. третий ресурс расходуется не полностью и остаётся в количестве 4 единиц. Этот ресурс — недефицитный, его оценка $y*¿_3=0$.

16. Двойственный симплекс-метод.

Для двойственного симплекс-метода можно использовать формат симплексных таблиц, описанный для прямого симплекс-метода. Однако в них среди коэффициентов b_i, $i=1,\dots,m$ могут быть отрицательные (если таблица не является оптимальной), а все коэффициенты c_j неотрицательны, т.е. выполнен признак оптимальности прямого симплекс-метода.

В ДСМ для получения текущего решения мы также полагаем все небазисные переменные равными нулю, а значения базисных переменных — равными a_{i0}, $i=1,\dots,m$. Далее мы находим ведущий элемент, причем сначала ищем ведущую строку, а затем — ведущий столбец. В отличие от прямого симплекс-метода здесь ведущий элемент является отрицательным. Формулы преобразования таблиц остаются без изменений.

Таким образом, в этом методе также происходит перебор совокупностей базисных переменных. Двойственный симплекс-метод можно рассматривать как прямой метод решения двойственной задачи ЛП. Ниже для ДСМ мы будем использовать несколько другой формат симплексной таблицы, поскольку он является более удобным при решении задач целочисленного линейного программирования алгоритмами отсечения.

Задача ЛП в приведенной форме:

$$
\begin{array}{l}
f(x) = c_0 - c_1x_1 - \cdots - c_nx_n \to \max \\
\qquad a_{11}x_1 + \cdots + a_{1n}x_n + \quad x_{n+1} \qquad = b_1, \\
\qquad a_{21}x_1 + \cdots + a_{2n}x_n + \qquad x_{n+2} \qquad = b_2, \\
\qquad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
\qquad a_{m1}x_1 + \cdots + a_{mn}x_n + \qquad\quad x_{n+m} = b_m, \\
\qquad x_j \geq 0, j = 1, \dots, n+m.
\end{array}
$$

Начальная симплексная таблица для ДСМ:

		x_1	$\cdots$	x_q	$\cdots$	x_n
f	c_0	c_1	$\cdots$	c_q	$\cdots$	c_n
x_1	0	-1	0	0	0	0
$\cdots$	$\cdot$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
x_n	0	0	0	0	0	-1
x_{n+1}	b_1	a_{11}	$\cdots$	a_{1q}	$\cdots$	a_{1n}
x_{n+2}	b_2	a_{21}	$\cdots$	a_{2q}	$\cdots$	a_{2n}
$\cdots$	$\cdot$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
x_{n+p}	b_p	a_{p1}	$\cdots$	a_{pq}	$\cdots$	a_{pn}
$\cdots$	$\cdot$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
x_{n+m}	b_m	a_{m1}	$\cdots$	a_{mq}	$\cdots$	a_{mn}

c_j в таблицу берутся также с противоположным знаком.

Здесь переменные $x_1,\dots,x_n$ — небазисные, а переменные $x_n+1,\dots,x_n+m$ — базисные. Заметив, что небазисным переменным соответствуют строки, которые содержат один ненулевой элемент (-1). Будет предполагать, что $c_j \geq 0$, $j=1,\dots,n$. В этом случае симплексная таблица называется двойственно допустимой.

Шаги:

1. Составить начальную допустимую симплексную таблицу.
2. Если $b_i \geq 0$ для любого $i=1,\ldots,n+m$, то найдено оптимальное решение $x*¿$ и алгоритм прекращает работу. При этом значения переменных в $x*¿$ равны соответствующим значениям b_i, $i=1,\ldots,n+m$.
3. Если существует строка с номером $p \geq 1$ такая, что $b_p<0$ и $a_{pj} \geq 0$ для любого j=1,…,n, то допустимое множество задачи пусто, следовательно задача неразрешима и алгоритм прекращает работу.
4. Строка с номером p>=1 выбирается в качестве ведущей если $b_p \leq 0$ (если таких несколько, то выбираем любая).
5. Столбец с номером q выбирается в качестве ведущего, если отношение c_j / a_{pj} максимальное среди столбцов, а $a_{pj}<0$.
6. Новые элементы ведущего столбца вычисляются делением на -ведущий элемент $(-a_{pq})$. Остальные новые элементы по правилу прямоугольника (старый минус произведение соседних угловых, делённых на ведущий). И затем по-новой на шаг 2.

Пример построения:

$$
\begin{aligned}
f(x) = -2x_1 - 8x_2 - x_3 - 5x_4 &\to max \\
-2x_1 + x_2 - 3x_3 + x_4 &\geq 18, \\
3x_1 + 4x_2 + 2x_3 - 3x_4 &\geq 30, \\
x_1 + 2x_2 + 4x_3 + 2x_4 &\geq 24, \\
x_1, x_2, x_3, x_4 &\geq 0.
\end{aligned}
$$

$$
\begin{aligned}
f(x) = -2x_1 - 8x_2 - x_3 - 5x_4 &\to max \\
-2x_1 + x_2 - 3x_3 + x_4 - x_5 \qquad &= 18, \\
3x_1 + 4x_2 + 2x_3 - 3x_4 \quad - x_6 \quad &= 30, \\
x_1 + 2x_2 + 4x_3 + 2x_4 \qquad - x_7 &= 24, \\
x_1, x_2, x_3, x_4, x_5, x_6, x_7 &\geq 0.
\end{aligned}
$$

$$
\begin{aligned}
f(x) = -2x_1 - 8x_2 - x_3 - 5x_4 &\to max \\
2x_1 - x_2 + 3x_3 - x_4 + x_5 \qquad &= -18, \\
-3x_1 - 4x_2 - 2x_3 + 3x_4 \quad + x_6 \quad &= -30, \\
-x_1 - 2x_2 - 4x_3 - 2x_4 \qquad + x_7 &= -24, \\
x_1, x_2, x_3, x_4, x_5, x_6, x_7 &\geq 0.
\end{aligned}
$$

		x_1	x_2	x_3	x_4 ↓
f	0	2	8	1	5
x_1	0	-1	0	0	0
x_2	0	0	-1	0	0
x_3	0	0	0	-1	0
x_4	0	0	0	0	-1
→ x_5	-18	2	-1	3	-1^*
x_6	-30	-3	-4	-2	3
x_7	-24	-1	-2	-4	-2

17. Транспортная задача.

Классическая транспортная задача может быть сформулирована следующим образом. Имеется m пунктов производства (хранения) и n пунктов потребления однородного продукта. Известны объемы производства (запасы) a_i поставщика A_i, $i=1,\ldots,m$, потребности (спрос) b_j потребителя B_j, $j=1,\ldots,n$, а также стоимость перевозки (транспортные затраты) единицы объёма продукции для каждой пары поставщик-потребитель c_{ij}, $i=1,\ldots,m$, $j=1,\ldots,n$. Требуется составить такой план перевозок, при котором потребности всех потребителей будут удовлетворены за счёт имеющихся запасов продукта в пунктах производства и общие транспортные расходы по доставке продукта будут минимальными.

Исходные данные обычно представляются в виде таблицы:

	B_1	...	B_j	...	B_n	
A_1	c_{11}	...	c_{1j}	...	c_{1n}	a_1
...	...	...	...	...	...	...
A_i	c_{i1}	...	c_{ij}	...	c_{in}	a_i
...	...	...	...	...	...	...
A_m	c_{m1}	...	c_{mj}	...	c_{mn}	a_m
	b_1	...	b_j	...	b_n	

Составим математическую модель ТЗ. Для этого введём переменные x_{ij}, $i=1,\ldots,m$, $j=1,\ldots,n$, которые обозначают количество продукта, перевозимого от $i-го$ поставщика к $j-му$ потребителю. Матрица

$$x = \begin{pmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{in} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{m1} & \cdots & x_{mj} & \cdots & x_{mn} \end{pmatrix}$$

называется планом перевозок. Общая стоимость перевозок f при плане x будет равна $f(x)=\sum_{(i=1)}^{m}\sum_{(j=1)}^{n} c_{ij}x_{ij}$.

При этом на план перевозок накладываются следующие ограничения. Для каждого поставщика суммарный объем вывезенного продукта не должен превышать объем производства (запаса) этого продукта, а потребность в продукте каждого потребителя должна быть полностью удовлетворена за счет перевозок от всех поставщиков. Первому требованию соответствуют неравенства $\sum_{(j=1)}^{n} xij \le a_i$, $i=1,\ldots,m$, а второму неравенства вида $\sum_{(i=1)}^{m} xij \ge b_j$, $j=1,\ldots,n$. Кроме того, по смыслу задачи все переменные x_{ij} должны быть неотрицательны.

Таким образом, получаем следующую математическую модель ТЗ:

$$f(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \rightarrow min,$$

$$\sum_{j=1}^{n} x_{ij} \le a_i, \ i = 1, ..., m,$$

$$\sum_{i=1}^{m} x_{ij} \ge b_j, \ j = 1, ..., n,$$

$$x_{ij} \ge 0, \ i = 1, ..., m; \ j = 1, ..., n.$$

Транспортная задача разрешима тогда и только тогда, когда выполнено следующее соотношение $\sum_{(j=1)}^{n} b_j \le \sum_{(i=1)}^{m} a_i$ (суммарные потребности меньше меньше или равны суммарным запасам). Если условие выполняется как равенство, то ТЗ сбалансированная, в противном случае — несбалансированная.

Несбалансированную ТЗ можно привести к сбалансированной введением фиктивного потребителя ($n+1$), у которого стоимости перевозок равны 0.

Решение ТЗ разбивается на три этапа. На первом этапе ищется какой-либо начальный допустимый план перевозок. На втором этапе найденный план проверяется на оптимальность. На третьем этапе (если план неоптимальный) определяется новый допустимый план перевозок.

18. Методы нахождения начального плана перевозок в транспортной задаче.

Метод северо-западного угла. Идёт от самой первой клетки таблицы. Есть три случая:

1. Если $a_1 < b_1$, то положим $x_{11} = a_1$. Следовательно первый поставщик перевёз весь продукт первому потребителю, и его запас равен нулю, поэтому $x12 = x13 = ... = x1m = 0$. Неудовлетворенный спрос первого потребителя равен $b'_1 = b_1 - a_1$.
2. Если $a_1 > b_1$, то положим $x_{11} = b_1$. Таким образом, спрос первого потребителя полностью удовлетворён, $x_{21} = x_{31} = ... = x_{m1} = 0$, а остаток продукта в первом пункте производства равен $a'_1 = a_1 - b_1$.
3. В случае $a_1 = b_1$ из рассмотрения можно исключить и поставщика, и потребителя. Однако для дальнейшего решения задачи методом потенциалов необходимо исключить только поставщика либо только потребителя. Например, если исключается поставщик, то спрос потребителя считается неудовлетворенным и равным нулю.

Далее рассматривается северо-западный угол оставшейся незаполненной части таблицы и повторяются те же действия. В результате получится начальный план перевозок.

Пример:

	B_1	B_2	B_3	B_4	
A_1	5	7	1	4	50
A_2	11	3	7	9	60
A_3	2	4	8	3	10
	20	40	30	30	

	B_1	B_2	B_3	B_4	a'_i			
A_1	20	30			50	30	0	0
A_2		10	30	20	60	50	20	0
A_3				10	10	0	0	0
b'_j	20	40	30	30				
	0	10	0	10				
	0	0	0	0				

Суммарная стоимость перевозок по данному плану равна $f(x)=20*5+30*7+10*3+30*7+20*9+10*3=760$

Метод минимальной стоимости. Отличие от предыдущего метода только в том, что выбирается клетка с минимальной стоимостью перевозки.

Пример:

	B_1	B_2	B_3	B_4	a_i'			
A_1			30	20	50	20	0	0
A_2	10	40		10	60	20	10	0
A_3	10				10	0	0	0
b_j'	20	40	30	30				
	10	0	0	10				
	0	0	0	0				

19. Метод потенциалов.

Введём переменные двойственной задачи u_i, $i=1,\ldots,m$ и v_j, $j=1,\ldots,n$, которые соответствуют поставщикам и потребителям. Они называются потенциалами или платежами.

1. Для текущего плана перевозок найти значения переменных u_i, v_j из системы уравнений $v_j — u_i = c_{ij}$, где c_{ij} клетки распределенных перевозок из начального плана перевозок (базисные клетки).
2. Пусть $delta_{ij} = v_j - u_j - c_{ij}$. Если выполняется $delta_{ij} \leq 0$ для всех перевозок, то текущий план оптимален. Вычислять данные значения надо только для небазисных (незаполненных) клеток.
3. Для построения нового плана перевозок найти клетку транспортной таблицы, в которой $delta_{ij} > 0$ (можно максимальную выбирать). Построить цикл, одна вершина которого находится в найденной (небазисной) клетке, а остальные — в базисных клетках. Циклом называются несколько клеток, соединенных замкнутой ломаной линией, которая в каждой клетке цикла совершает поворот на 90 градусов. Клетку с $delta_{ij} > 0$ отметить знаком $+¿$. Соседние с ней клетки (вершины) — знаком $-¿$. Далее знаки в вершинах чередуются. В вершинах помеченных знаком -, выбрать минимальный объем перевозок и перераспределить его по циклу, т.е. там где был минус этот минимальный объём отнять, а там где плюс — прибавить. Перейти на шаг 1.

Пример:

	B_1	B_2	B_3	B_4
A_1			30	20
A_2	10	40		10
A_3	10			

	B_1	B_2	B_3	B_4	
A_1	5	7	1	4	50
A_2	11	3	7	9	60
A_3	2	4	8	3	10
	20	40	30	30	

$$
\begin{aligned}
v_3 - u_1 &= 1,\\
v_4 - u_1 &= 4,\\
v_1 - u_2 &= 11,\\
v_2 - u_2 &= 3,\\
v_4 - u_2 &= 9,\\
v_1 - u_3 &= 2.
\end{aligned}
$$

$$u_1 = 5;\ u_2 = 0;\ u_3 = 9;\ v_1 = 11;\ v_2 = 3;\ v_3 = 6;\ v_4 = 9.$$

$$\begin{aligned}
\Delta_{11} &= v_1 - u_1 - c_{11} = 11 - 5 - 5 = \quad 1 > 0,\\
\Delta_{12} &= v_2 - u_1 - c_{12} = \ 3 - 5 - 7 = \ -9 < 0,\\
\Delta_{11} &= v_3 - u_2 - c_{23} = \ 6 - 0 - 7 = \ -1 < 0,\\
\Delta_{11} &= v_2 - u_3 - c_{32} = \ 3 - 9 - 4 = -10 < 0,\\
\Delta_{11} &= v_3 - u_3 - c_{33} = \ 6 - 9 - 8 = -11 < 0,\\
\Delta_{11} &= v_4 - u_3 - c_{34} = \ 9 - 9 - 3 = \ -3 < 0.
\end{aligned}$$

$$(1,1) - (1,4) - (2,4) - (2,1) - (1,1).$$

	B_1	B_2	B_3	B_4
A_1	$^+$		30	20^-
A_2	10^-	40		10^+
A_3	10			

	B_1	B_2	B_3	B_4
A_1	10		30	10
A_2		40		20
A_3	10			

Стоимость перевозок по данному плану равна $f(x^2)=10*5+30*1+10*4+40*3+20*9+10*2=440$. Далее выполняются следующие итерации данного метода, пока текущий план перевозок не станет оптимальным.

Теорема. Если транспортная задача является сбалансированной и невырожденной, то метод потенциалов за конечное число шагов находит её оптимальное решение.

20. Задача о назначениях.

Задача о назначениях представляет собой специальный тип транспортной задачи ЛП, суть которой заключается в поиске наиболее выгодного соответствия между равным количеством исполнителей и работ. Основная идея строится на жестком принципе единоразового распределения: за каждым исполнителем закрепляется строго одна задача, и каждая задача выполняется строго одним исполнителем. Математическая модель такой задачи всегда описывается квадратной матрицей, в ячейках которой указаны затраты (или эффективность) для каждой возможной пары, а целью является минимизация общей суммы издержек или максимизация совокупного результата. Поскольку в данной задаче каждый ресурс используется полностью и однократно, искомые переменные принимают только бинарные значения (1, 0), что превращает её в задачу выбора оптимального набора из n ячеек в таблице размерности $n\,X\,n$.

21. Предмет и задачи теории игр.

Теория игр — это особый раздел исследования операций, изучающий математические модели ситуаций, в которых принятие решения зависит от нескольких сторон. Важную роль в этих ситуациях играют конфликты и совместные действия. Под конфликтом будем понимать любое явление, для которого можно определить, кто и как в этом явлении участвует, каковы возможные исходы этого явления, кто в каких исходах заинтересован и в чем состоит эта заинтересованность.

В теории игр конфликт называют игрой, а участников конфликта — игроками. Решения, которые могут принимать игроки, называются стратегиями. Каждый игрок располагает конечным или бесконечным числом стратегий. Ситуацией называется комбинация выбранных игроками стратегий. Выигрыш каждого игрока определяется его платежной функцией, значения которой зависят от сложившейся ситуации в игре.

Задача теории игр состоит в установлении принципов оптимального поведения игроков, в доказательстве существования оптимальных ситуаций, которые складываются в результате применения этих принципов, и разработке методов нахождения таких ситуаций.

22. Классификация игр.

Игры классифицируются по числу игроков (парные и игры n лиц (при $n>2$ в играх могут создаваться коалиции)), по числу стратегий (конечные и бесконечные), по свойствам платежной функции (игра с нулевой суммой, если общая сумма выигрышей всех игроков равна нулю (в парной игре с нулевой суммой выигрыш одного игрока равен проигрышу второго, такие игры называются антагонистическими)).

23. Развернутая и нормальная форма игры.

Нормальная форма игры (часто называемая матричной) представляет собой максимально упрощенное описание, где основной акцент сделан на финальном результате. В этой форме предполагается, что игроки выбирают свои стратегии одновременно и независимо друг от друга. Основным инструментом здесь служит платежная матрица: строки соответствуют стратегиям первого игрока, а столбцы — стратегиям второго, а в ячейках на пересечении указаны выигрыши обоих участников. Такая форма идеально подходит для анализа статических игр (например камень-ножницы-бумага), где не важна последовательность действий, а важен лишь итоговый набор выбранных стратегий.

Развернутая форма игры (или позиционная) описывает процесс взаимодействия гораздо подробнее, представляя его в виде игрового дерева. Этот формат визуализирует игру как последовательность ходов во времени. В нем четко зафиксированы вершины (узлы), в которых игроки принимают решения, ветви, обозначающие возможные действия, и информационные множества, которые показывают, что именно игрок знает о предыдущих ходах оппонента в данный момент. Развернутая форма незаменима для анализа динамических, многошаговых процессов (например, шахмат или переговоров), т.к. она позволяет учитывать очередность ходов и реакцию игроков на действия друг друга.

24. Матричные игры.

Конечные антагонистические игры называются матричными. Данное название связано с формой записи таких игр. Так как число стратегий каждого игрока в матричной игре конечно, то их можно пронумеровать. Пусть число стратегий игрока I равно m, а число стратегий игрока II — n. Далее будем называть их чистыми стратегиями.

Платежные функции (функции выигрыша) игроков в матричной игре могут быть заданы в виде платежной матрицы $A=(a_{ij})$ размера $m X n$, где a_{ij} — выигрыш игрока I (и соотв. проигрыш игрока II) в ситуции (i,j), т.е. тогда когда первый игрок выберет стратегию i, а второй игрок — стратегию j.

В задачах принятия решений критерий оптимальности в значительной степени определяется информацией, которой располагает «лицо, принимающее решение». Матричные игры представляют собой случай полного отсутствия информации о действиях другого игрока, поэтому в таких играх, как правило, используют наиболее пессимистичный минимаксный критерий или, другими словами, принцип гарантированного результата.

25. Принцип минимакса.

Итак, пусть игроки I и II участвуют в матричной игре с платежной матрицей $A=(a_{ij})$ размера $m X n$. Выигрыш игрока I или, что то же самое, проигрыш игрока II в ситуации (i,j) равен a_{ij}. Очевидно, что игрок I будет стараться увеличить свой выигрыш, в то время как второй игрок будет пытаться его уменьшить.

Если игрок I выбрал стратегию i, то второй, будучи разумным игроков, должен ответить такой стратегией j, чтобы выигрыш a_{ij} был минимальным среди чисел $a_{i1},\ldots,a_{(in)}$. Обозначим этот выигрыш u_i, т.е. $u_i = min_j a_{ij}$. Величина u_i есть наименьший, т.е. гарантированный выигрыш игрока I при выборе им стратегии i безотносительно к решениям второго игрока. Следовательно, игроку I выгодно выбрать ту стратегию, которая даёт максимальное значение u_i: $u = max_i u_i = max_i min_i a_{ij}$. Величина u называется нижней ценой игры в чистых стратегиях, а стратегия игрока I, на которой достигается это величина, - максиминной стратегией.

Аналогично получаем, что величина $v_j = max_i a_{ij}$ есть максимальный, т.е. гарантированный проигрыш игрока II при выборе им стратегии j. Игроку II из всех стратиегий выгодно выбрать ту, при которой его гарантированный проигрыш v_j минимален: $v = min_j v_j = min_j max_i a_{ij}$. Величина v называется верхней ценой игры в чистых стратегиях, а стратегия игрока II, на которой достигается эта величина, - минимаксной стратегией.

Ещё раз отметим, что, выбрав максиминную стратегию, игрок I гарантированно получит выигрыш, не меньший u, а игрок II, выбрав минимаксную стратегию, не проиграет больше v.

Приведенные рассуждения носят названия принципа минимакса (или принципа гарантированного результата). Кратко этот принцип может быть сформулирован следующим образом: каждый игрок стремится максимально увеличить свой гарантированный выигрыш.

В теории игр оптимальность решения связывают с ситуаций, в которой ни одному игроку не выгодно изменять свою стратегию. В этом случае говорят, что имеет место ситуация равновесия, а соответствующие стратегии являются устойчивыми. При этом выполняется равенство $u=v$.

Ситуация равновесия непосредственно связана с понятием седловой точки платежной матрицы. Элемент a_{pq} матрицы A называется седловой точкой, если $a_{iq} \le a_{pq} \le a_{pj}$ для всех $i=1,\dots,m$, $j=1,\dots,n$. Другими словами, элемент a_{pq} является одновременно минимальным в строке и максимальным в столбце.

Теорема. Для любой матричной игры равенство $u=v$ имеет место тогда и только тогда, когда платежная матрица обладает седловой точкой.

Таким образом, седловая точка платежной матрицы дает ситуацию равновесия (p,q) в матричной игре, при которой если один из игроков придерживается своей минимаксной (максиминной) стратегии, то другой игрок не может улучшить свое положение, отступая от своей максиминной (минимаксной) стратегии. Заметим, что седловая точка может быть не единственной.

26. Чистые и смешанные стратегии.

Если (p,q) — седловая точка, то стратегии p, q называются оптимальными чистыми стратегиями, ситуация (p,q) — решением игры в чистых стратегиях, а сама матричная игра — разрешимой в чистых стратегиях. Величина $u=v$ называется в этом случае чисто ценой игры.

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

Здесь элемент $a_{23}=1$ является единственной седловой точкой матрицы A, следовательно, игра разрешима. Оптимальные стратегии — это вторая для первого игрока и третья — для второго. Цена игры равна 1.

В случае, когда матрица A не имеет седловой точки, ситуации равновесия в игре нет и она не разрешима в чистых стратегиях. В таких играх игроки должны выбирать свои чистые стратегии с некоторой вероятностью. При этом вероятности выбора стратегий определяются таким образом, чтобы средний выигрыш (проигрыш) был максимальным (минимальным).

Смешанной стратегией игрока в матричной игре называется вектор вероятностей выбора его чистых стратегий. Так если $i=1,\dots,m$ — чистые стратегии игрока I, а $j=1,\dots,n$ — чистые стратегии игрока II, то смешанные стратегии игроков I и II — это векторы вероятностей $x=(x1,\dots,x_m)$ и $y=(y_1,\dots,y_n)$ соответственно, где x_i — вероятность выбора игроком I чистой стратегии i,

$i=1,\ldots,m$, а y_i — вероятность выбора игроком II чистой стратегии j, $j=1,\ldots,n$. Очевидно, векторы x и y должны удовлетворять условиям:

$\sum_{(i=1)}^{m} x_i=1,\ x_i\geq 0,\ i=1,\ldots,m.$

$\sum_{(j=1)}^{n} y_i=1,\ y_i\geq 0,\ j=1,\ldots,n.$

Величины $u*¿$ (выигрыш первого игрока) и $v*¿$ (проигрыш второго игрока) называются соответственно нижней и верхней ценой игры в смешанных стратегиях.

Матричная игра называется разрешимой в смешанных стратегиях, если $u*¿=v*¿$. При этом стратегии $x*¿$, $y*¿$, для которых $u(x*¿)=u*¿=v*¿=v(y*¿)$, называются оптимальными смешанными стратегиями, а величина $u*¿v*¿$ - ценой игры в смешанных стратегиях. Оптимальные смешанные стратегии образуют ситуацию равновесия в смешанных стратегиях.

Оптимальные смешанные стратегии игроков, а также цена игры в смешанных стратегиях могут быть найдены при решении пары взаимно двойственных задач линейного программирования.

Пример построения задачи

:

	B1	B2
A1	4	10
A2	9	0

$U \rightarrow max$

$U-4*x_1-9*x_2\leq 0$

$U-10*x_1\leq 0$

$x_1+x_2=1$

$x_1,x_2,U\geq 0$

$V \rightarrow min$

$V-4*y_1-10*x_2\leq 0$

$V-9*y_1\leq 0$

$y_1+y_2=1$

$y_1,y_2,V\geq 0$

27. Теорема Фон Неймана.

Теорема. (О минимаксе) В любой матричной игре имеет место равенство $u*¿=v*¿$, т.е. любая матричная игра разрешима в смешанных стратегиях.

28. Задача целочисленного линейного программирования.

Это задача ЛП с дополнительным требованием целочисленности значений переменных. Рассмотрим следующую постановку задачи ЦЛП:

$f(x)=\sum_{(j=1)}^{n} c_j x_j \rightarrow max$

$\sum_{(j=1)}^{n} a_{ij} x_j \leq b_i,\ i=1,\ldots,m$

$x_j \geq 0$, x_j принадлежит Z, $j=1,\ldots,n$, Z — множество целых чисел.

Такая задача называется линейной релаксацией задачи ЦЛП, а множество, определяемое ограничениями — релаксационным множеством.

Частным случаем задачи ЦЛП является задача булева линейного программирования, которая отличается от задачи выше последним условием. Вместо требования целочисленности в задачах булева линейного программирования на переменные накладываются ограничения x_j принадлежащее $0{,}1$, $j=1,\ldots,n$. Такие переменные называются булевыми.

29. Методы решения задачи целочисленного линейного программирования.

1. Метод округления. Сначала находится оптимальное решение $x*¿$ линейной релаксации. Далее значения переменных, не являющихся целыми в $x*¿$, округляют так, чтобы получить решение z' с целочисленными значениями.
2. Метод ветвей и границ. Сначала находится оптимальное решение $x*¿$ линейной релаксации. Если есть дробное значение какой либо из переменных, то выбираем её и строим ветвление (например для $x*¿_2=3{,}7$ новые ограничения для ветвления будут $x_2 \leq 3$ и $x_2 \geq 4$). Находятся два новых оптимальных решения с добавленным ограничением из прошлого шага. Если какое-то из них целочисленное, то конец. В ином случае добавляется новое ограничение для каждой ветви и т.д. и т.п. цикл продолжается.
3. Метод отсечения Гомори. Решается просто задача ЛП двойственным симплекс-методом. Если решение не целочисленное, выбирается строка с самой большой дробная часть в решении $x*¿$ и строится новое ограничение: дробные части переменных в этой строке умноженные на сами переменные (кроме той переменной, которой эта строка принадлежит) <= дробной части свободного члена. Ограничение добавляется в таблицу и задача снова считается ДСМ. Ограничения добавляются пока решение не будет целочисленным.

30. Метод ветвей и границ.

Сначала находится оптимальное решение $x*¿$ линейной релаксации. Если есть дробное значение какой либо из переменных, то выбираем её и строим ветвление (например для $x*¿_2=3,7$ новые ограничения для ветвления будут $x_2 \leq 3$ и $x_2 \geq 4$). Находятся два новых оптимальных решения с добавленным ограничением из прошлого шага. Если какое-то из них целочисленное, то конец. В ином случае добавляется новое ограничение для каждой ветви и т.д. и т.п. цикл продолжается.

31. Схема Лэнд и Дойг.

Основная идея заключается в том, чтобы не перебирать все возможные комбинации целых чисел, а последовательно «сужать кольцо» вокруг оптимального решения, отбрасывая целые пласты вариантов, которые заведомо хуже уже найденных.

1. Сначала задача решается как обычная задача ЛП (без учёта целочисленности). Если решение $x*¿$ оказалось целым — задача решена.
2. Если переменная x_j должна быть целой, но в решении $x*¿$ она получилась дробной (например $x_j=4,3$), то эта переменная выбирается для ветвления.
3. Создаются две новые подзадачи. К ограничениям исходной задачи добавляются условия (интервал между 4 и 5 исключается, т.к. там нет целых чисел для решения):
    1. В левой ветви $x_j \leq 4$ (целая часть числа);
    2. В правой ветви $x_j \geq 5$ (целая часть $+1$).
4. Для каждой ветви решается своя задача ЛП. Значение её целевой функции f становится «верхней границей» (для задачи на максимум). Если в какой-то ветви f меньше, чем уже найденное нами любое целое решение (рекорд) эту ветвь «отсекают» и больше не рассматривают. Изначально рекорд равен $-\infty$.

Метод ветвей и границ это общая концепция, а схема Лэнд и Дойг — конкретный метод решения.

32. Задача о рюкзаке.

Есть задача рюкзаке, где предметы в рюкзак можно поместить лишь один раз. Найти: совокупность предметов с максимальной суммарной полезностью, которая помещалась бы в заданный объём.

Математическая модель задачи булевого одномерного рюкзака:

$f(x)=c_1 x_1+c_2 x_2+\ldots+c_n x_n \rightarrow max$

$a_1 x_1+a_2 x_2+\ldots+a_n x_n \leq b$

$x_j \, принадлежит (0,1), \; j=1,\ldots,n$

А есть задача на одномерный целочисленный рюкзак, где предметы можно складывать бесконечно много раз для одного типа. Найти: количество предметов каждого типа максимальной суммарной полезности, которые помещаются в рюкзак заданного объема.

Математическая модель одномерного целочисленного рюкзака:

$f(x)=c_1x_1+c_2x_2+\ldots+c_nx_n \to max$

$a_1x_1+a_2x_2+\ldots+a_nx_n \leq b$

$x_j \, принадлежит \, Z, \; j=1,\ldots,n$

33. Уравнение Беллмана.

Принцип оптимальности Беллмана. Каким бы ни было начальное состояние и решение, принятое в начале, последующие решения должны быть оптимальными по отношению к состоянию, возникшему в результате первого шага.

Уравнение:

$f_k(y)=max_{(x_k \, принадлежит \, Z)}(c_kx_k+f_{(k+1)}(y-a_kx_k)),$

где $f_k(y)$ — оптимальное значение целевой функции при размещении предметов k, $k+1$, …, n в рюкзаке объёма y.

34. Схема динамического программирования для задачи о рюкзаке.

ДП решает задачу, разбивая её на поиск лучшего решения для всех промежуточных весов — от 0 до максимальной вместимости и постепенно добавляя предметы в подсчёты.

У нас есть:

1. n — количество типов предметов
2. b — общая вместимость.
3. a_i — вес предмета типа i.
4. c_i — стоимость предмета типа i.
5. $f_k(y)$ — оптимальное значение целевой функции при размещении предметов k, $k+1$, …, n в рюкзаке объёма y.
6. $f_1(b)$ — оптимальное значение целевой функции исходной задачи.

По уравнению Беллмана: $f_k(y)=max_{(x_k \, принадлежит \, Z)}(c_kx_k+f_{(k+1)}(y-a_kx_k)),$ мы считаем для каждого объёма от 0 до b и для каждого количества предметов k от n до 1 (мы как бы вводим по одному предмету уменьшая значения k, т.е. в начале мы можем положить только последний предмет, потом добавляется предпоследний и т.д.).

35. Задача многокритериальной оптимизации.

В обычной задаче ЛП мы ищем максимум прибыли, либо минимум затрат. В многокритериальной оптимизации мы хотим и то, и другое одновременно.

Пример при покупке автомобиля. Критерии:

1. Цена (хочется меньше).
2. Комфорт (хочется больше).

Проблема в том, что при улучшении одного критерия (комфорта) почти всегда ухудшается другой (цена растет). В таких задачах не существует одного идеального решения, которое было бы лучшим по всем пунктам сразу.

Математически это записывается как векторная функция:

$F(x)=(f_1(x), f_2(x), \ldots, f_k(x)) \rightarrow min(max)$

36. Парето-оптимальные решения

Парето-оптимальность

$f(x) \rightarrow max_k \, k=1, \ldots, p$

x принадлежит D

Решение $x*¿$ из D называется Парето-оптимальным, если не существует такого x из D, что $f_k(x) \geq f_k(x*¿)$ для всех $k=1, \ldots, p$, причем хотя бы для одного k неравенство строгое.

Решение называется оптимальным по Парето, если его нельзя улучшить по какому-то одному критерию, не ухудшив его при этом хотя бы по одному из остальных.

Теорема. В любой многокритериальной задаче ЛП Парето-множество является объединением некоторых граней многогранника допустимых решений D (грани могут иметь различную размерность).

Теорема. Если множество допустимых решений D многокритериальной задачи ЛП не пусто, то множество парето-оптимальных вершин реберно-связано, т.е. для любых двух парето-оптимальных вершин существует связывающий их маршрут, проходящий ребрам многогранника D, причем все точки этого маршрута парето-оптимальны.

37. Метод последовательных уступок.

Идея: один из критериев выбирается в качестве целевой функции, а для других задаются некоторые предельные значения (нижние оценки).

Пример. Задача построить завод, в которой есть три критерия: f_1 — чистая прибыль, f_2 — срок строительства. Вместо того чтобы искать $f_1 \rightarrow max$ при $f_2 \rightarrow min$ (или наоборот) мы вводим чёткое ограничение. Например срок строительства не больше 2-х лет, т.е. $f_2 \leq 2$. А целевая функция получается $f_1 \rightarrow max$. Задача превращается в обычную задачу однокритериальной оптимизации.

38. Свертка критериев.

Идея: замена системы целевых функций одной функцией — компромиссной.

Требования к компромиссной функции:

1. Учёт относительной важности каждого критерия.
2. Приведение критериев к безразмерной форме.

Виды свёрток:

1. Линейная. $L(x)=\sum_{(k=1)}^{p} a_k f_k(x)$, где a_k — коэффициент относительной важности $k-го$ критерия, причем $a_k \geq 0$, $k=1,\ldots,p$, $a_1+\ldots+a_p=1$. Теорема: для любой многокритериальной задачи ЛП линейная свёртка находит все Парето-оптимальные решения. Недостаток: сложность определения коэффициентов a_k.
2. Расстояние до «идеала». Мы представляем идеальную точку (которая обычно вне области допустимых решений) и пытаемся к ней максимально приблизиться своим решением в области допустимых решений.
    1. Ищется идеал для каждого критерия (максимум или минимум каждой функции). Из идеалов составляются координаты идеальной точки $L*¿=(f*¿_1,f*¿_2,\ldots,f*¿_k)$.
    2. Критерии нормируются, их приводят к единому масштабу (от 0 до 1).
    3. Для каждого варианта решения вычисляется расстояние до идеальной точки. Чаще всего используется Евклидово расстояние: $D=\sum_{(i=1)}^{k}(f_i(x)-f*¿_i)^2$
    4. Целевой функцией становится само расстояние D. Нам нужно найти такое решение x, при котором $D \rightarrow min$.
3. Взвешенный минимум. U(x) = min_(k=1..p) f_k (x) / f**_k → max, где f_k (x) — текущее значение данного критерия, f**_k — лучшее возможное значение этого критерия. Дробь показывает насколько мы далеко от идеала. Минимум в функции позволяет выбрать критерий, по которому самый низкий процент успеха (например, по прибыли достигли 90% от идеала, а по безопасности только 40%, вот безопасность и будем улучшать). А стремление к max позволяется ориентироваться на самый худший процент, т.к. в этом методе наша задача не максимизировать отдельный критерий, не обращая внимания на другие, а в купе все критерии подтянуть до достаточно оптимальных значений.