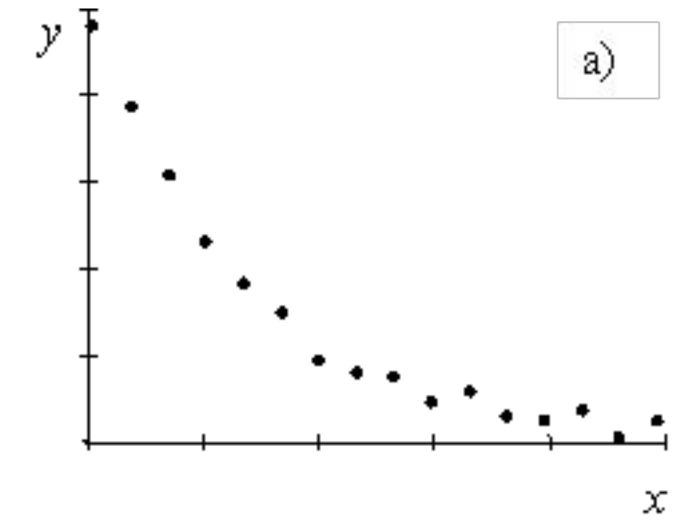
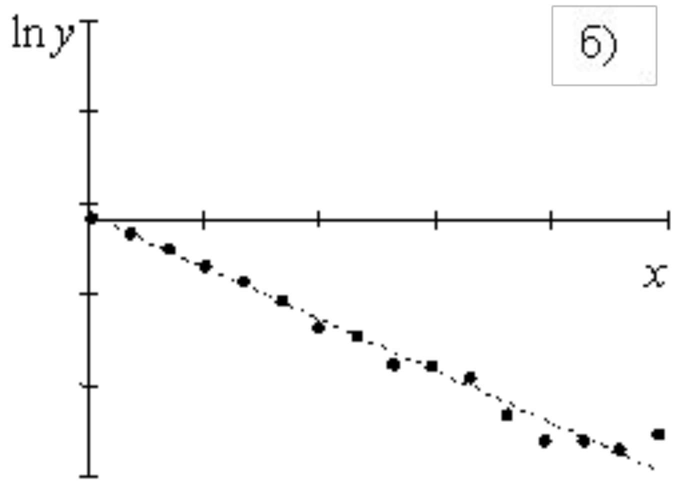
Аппроксимация

Аппроксимация – построение функции, усредняющей значения набора точек.

Аппроксимацию удобно применять в случае, когда экспериментальные данные получены с погрешностью (в этом случае интерполяция эффективна).

Задача аппроксимации:
Дан набор точек, и необходимо найти функцию (эмпирическую формулу), выражающую приближенную зависимость между точками.





Метод наименьших квадратов

$y = \varphi(x, c_0, c_1, \dots, c_m)$ Эмпирическая формула

$\varepsilon_i = \varphi(x_i, c_0, c_1, \dots, c_m) - y_i$ Отклонение эмпирической формулы от экспериментальных данных

$$Q = \sum_{i=0}^{n} (\varepsilon_i)^2 = \sum_{i=0}^{n} \left[\varphi(x_i, c_0, c_1, \dots, c_m) - y_i \right]^2$$

Среднеквадратичное отклонение эмпирической формулы от экспериментальных данных

Оптимальные параметры эмпирической функции находятся из соотношения:

$$\frac{\partial Q}{\partial c_0} = 0, \quad \frac{\partial Q}{\partial c_1} = 0, \quad \dots, \quad \frac{\partial Q}{\partial c_m} = 0$$

т.е.

$$\begin{cases} \sum_{i=0}^{n} [\varphi(x_i, c_0, c_1, \dots, c_m) - y_i] \left(\frac{\partial \varphi}{\partial c_0} \right)_i = 0 \\ \sum_{i=0}^{n} [\varphi(x_i, c_0, c_1, \dots, c_m) - y_i] \left(\frac{\partial \varphi}{\partial c_1} \right)_i = 0 \\ \dots\dots\dots \\ \sum_{i=0}^{n} [\varphi(x_i, c_0, c_1, \dots, c_m) - y_i] \left(\frac{\partial \varphi}{\partial c_m} \right)_i = 0 \end{cases}$$

где $\left(\frac{\partial \varphi}{\partial c_m} \right)_i = \frac{\partial}{\partial c_m} \varphi(x_i, c_0, c_1, \dots, c_m)$

Применение метода наименьших квадратов

$$\varphi(x,a,b) = a + bx$$

$$\begin{cases} \sum_{i=0}^{n}(a + bx_i - y_i) = 0 \\ \sum_{i=0}^{n}(a + bx_i - y_i)x_i = 0 \end{cases} \qquad \begin{cases} (n+1)a + b\sum_{i=0}^{n}x_i - \sum_{i=0}^{n}y_i = 0 \\ a\sum_{i=0}^{n}x_i + b\sum_{i=0}^{n}x_i^2 - \sum_{i=0}^{n}x_i y_i = 0 \end{cases}$$

Разделив все слагаемые в уравнениях (n+1) получим:

$$\begin{cases} a + b(x) - (y) = 0 \\ a(x) + b(x^2) - (xy) = 0 \end{cases}$$

Обозначим:

$$(x) = \frac{1}{n+1}\sum_{i=0}^{n}x_i \quad (x^2) = \frac{1}{n+1}\sum_{i=0}^{n}x_i^2$$

$$(y) = \frac{1}{n+1}\sum_{i=0}^{n}y_i \quad (xy) = \frac{1}{n+1}\sum_{i=0}^{n}x_i y_i$$

Получим:

$$a = (y) - b(x) \quad b = \frac{(xy) - (x)(y)}{(x^2) - (x)^2}$$

Исследование операций

Исследование операций (ИСО) — применение математических, количественных методов для обоснования решений во всех областях целенаправленной человеческой деятельности. Цель исследования операций — предварительное количественное обоснование оптимальных решений с опорой на показатель эффективности.

Исследование операций (ИСО) -
это наука о количественном обосновании оптимальных решений на основе построения и использования математической модели.

Основные понятия модели исследования операций

Целевая функция
Функция ограничения
Внутренние параметры
Оптимизационная модель (модель с целевой функцией)
Неоптимизационная модель (модель без целевой функцией)
Метод оптимизации (метод исследования операций)
Оптимальное решение

Разделы дисциплины исследования операций

Математическое программирование ("планирование") - методы отыскания экстремальных значений функции, на аргументы которой наложены ограничения.
Линейное программирование (ЛП) – задачи с линейными целевыми функциями и ограничениями.
Нелинейное программирование (НП) рассматривает задачи с нелинейными целевыми функциями и ограничениями.
Целочисленное линейное программирование – решение задач, у которых все или некоторые переменные должны принимать целочисленные значения.
Динамическое программирование выбор наиболее оптимального управления динамическим объектом во времени.
Аппарат **теории вероятностей** позволяет находить оптимальные параметры стохастических систем (СМО, регрессионный и корреляционный анализ и т.п.).
Теория игр и принятия решений рассматривает процессы выбора наилучшей из нескольких альтернатив в ситуациях определенности, в условиях риска (данные можно описать с помощью вероятностных распределений), в условиях неопределенности (вероятностное распределение либо неизвестно, либо не может быть определено).
Теория нечетких множеств позволяет производить оценку нечетких данных и нечетких суждений (например, с помощью этого аппарата можно оценивать субъективное мнение экспертов.

Одна задача линейного программирования

Задача:

Для окраски помещения необходимо купить 15 кг краски. Эту краску можно купить в банках двух типов: по 1,5 кг, стоимостью 10 рублей каждая, или банках весом 0,9 кг, стоимостью 8,5 рублей. Для перевозки используется ящик, в который может уместиться 8 банок первого типа или 25 банок второго типа.

Решение:

x_1 - количество банок первого типа (каждая банка занимает 1/8 ящика),
x_2 - количество банок второго типа (каждая банка занимает 1/25 ящика).

$1{,}5x_1 + 0{,}9x_2 >= 15$
$1/8x_1 + 1/25x_2 <= 1$
Стоимость покупки: $L=10x_1+8{,}5x_2 \text{ -> } \min$

$$\begin{cases} L = 10x_1 + 8{,}5x_2 \to \min; \\ 1{,}5x_1 + 0{,}9x_2 \geq 15; \\ \frac{1}{8}x_1 + \frac{1}{25}x_2 \leq 1. \end{cases}$$

Решение одной задачи линейного программирования графическим методом

$$\begin{cases} L = 10x_1 + 8,5x_2 \to \min; \\ 1,5x_1 + 0,9x_2 \ge 15; \\ \frac{1}{8}x_1 + \frac{1}{25}x_2 \le 1. \end{cases}$$

Возьмем, например, L=170, тогда
$170=10x_1+8,5x_2$
Передвигаем линию в направлении начала координат…

А- оптимальная точки
В – оптимальная точки
для целочисленных значений

Решение одной задачи линейного программирования

Найдем точку А из уравнения

$$\begin{cases} 1,5x_1 + 0,9x_2 = 15; \\ 1/8x_1 + 1/25x_2 = 1. \end{cases}$$

x1=5,8; x2=7,1; L3=118,35

Округление x1=6, x2=7 не является допустимым решением

Допустимым является решение x1=4, x2=10, L=125

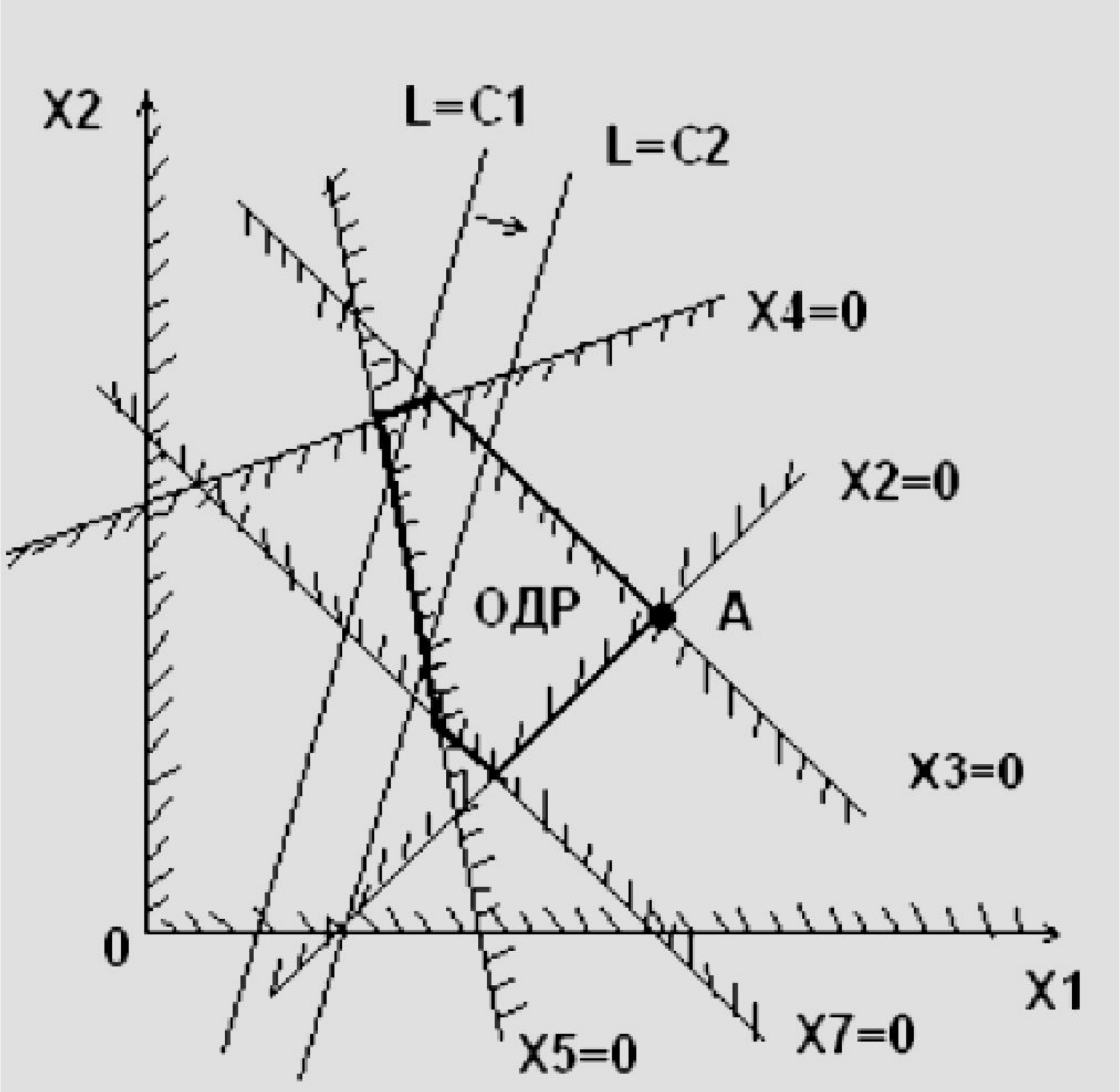
Еще одна задача линейного программирования

Имеются четыре вида продуктов П 1 , П 2 , П 3 , П 4 . Известна стоимость каждого продукта с 1 , с 2 , с 3 , с 4 . В рационе должно быть белков не менее b 1 ; жиров не менее b 2 ; углеводов не менее b 3 . Известно удельное содержание каждого вещества в единице продукта. Необходимо рассчитать рацион, содержащий необходимое количество полезных веществ и при этом, чтобы стоимость всего рациона была минимальной. Построим математическую модель:

j – номер продукта ; i – номер вещества;

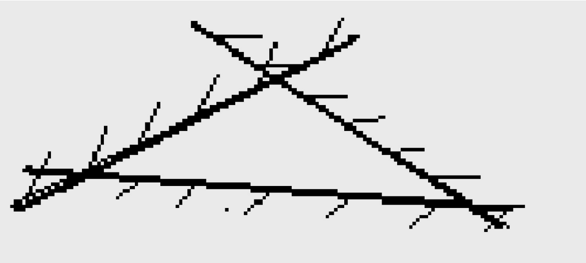
a_{ij} – удельное количество в j – м продукте i – го вещества

Геометрическая интерпретация задачи линейного программирования

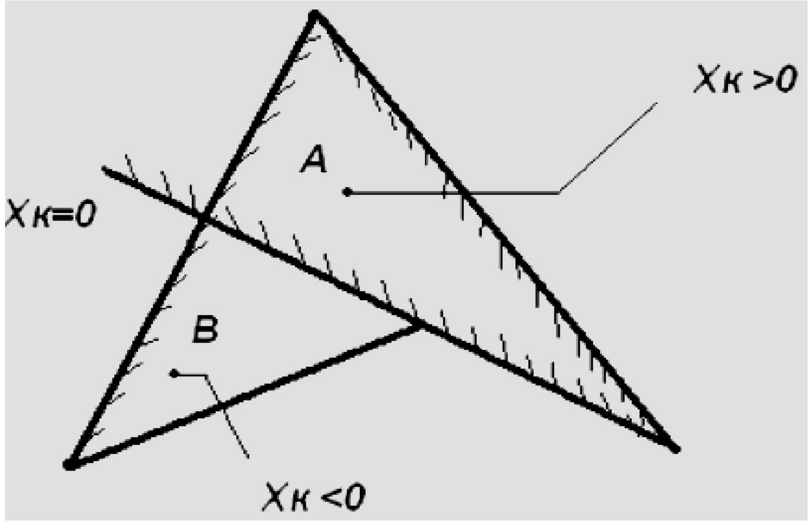


Варианты ОДЗ задачи линейного программирования

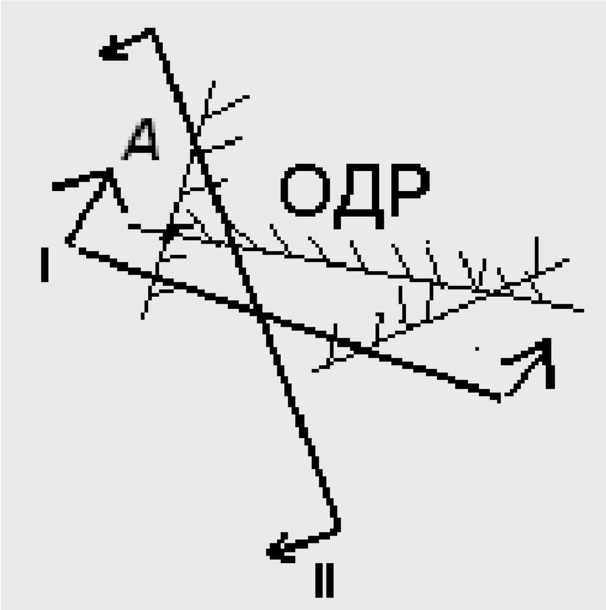
Нет решения



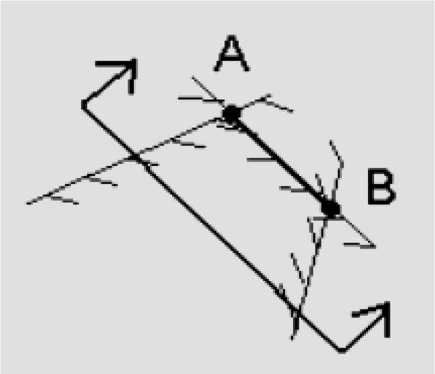
Выпуклый многогранник



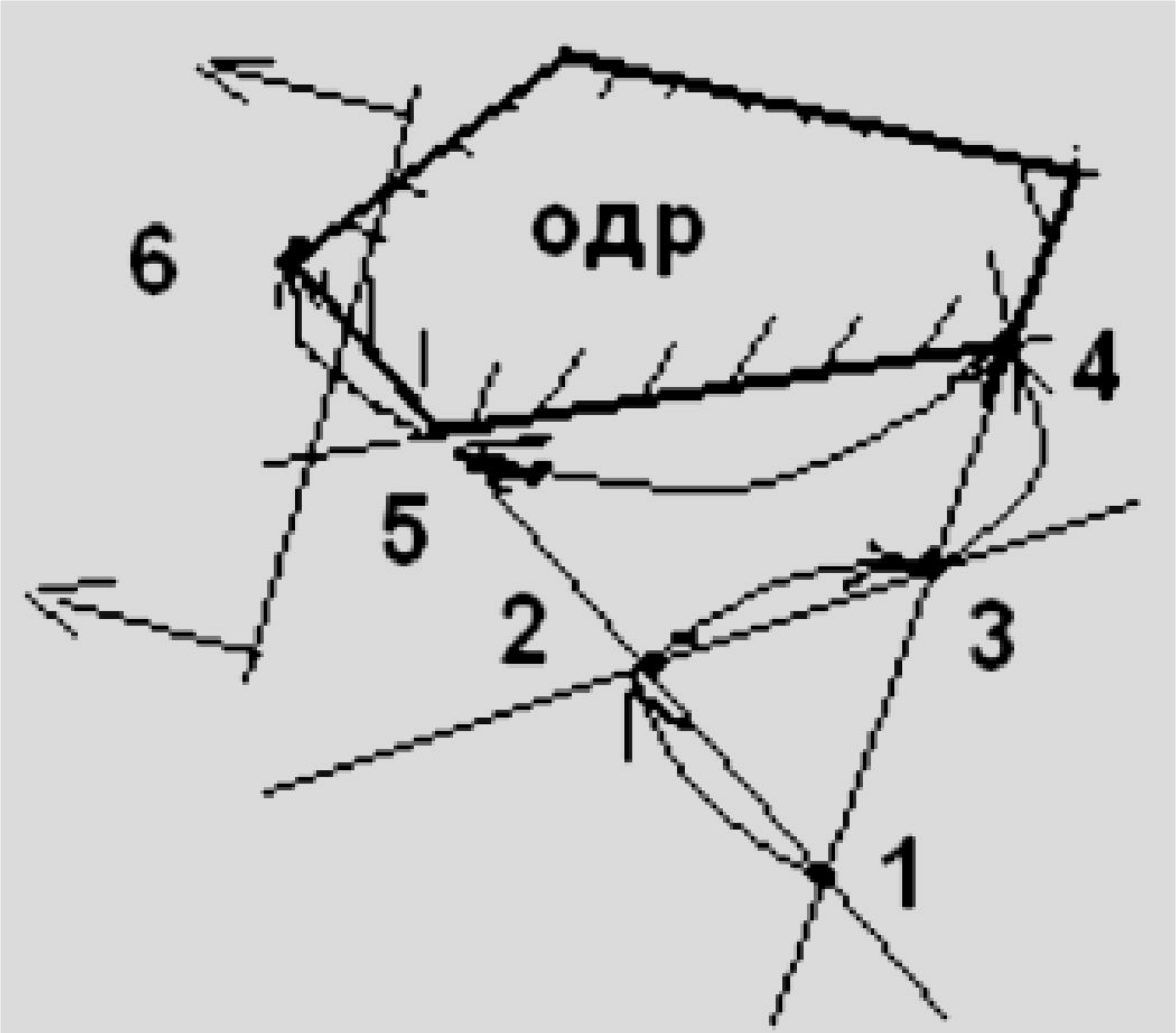
Открытая ОДР



Открытая ОДР



Симплекс-метод решения задачи ЛП



1,2,3 – недопустимые решения
4,5,6 – опорные точки.

Формы записи системы уравнений для ЛП

- Система уравнений и ограничений
- Каноническая.
- Матричная.

Система уравнений и ограничений

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

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

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

Таблица приёмов перевода системы управлений к каноническому виду

Название преобразования	До преобразования	После преобразования
Преобразование неравенства в равенство	$\sum_{j=1}^{n} a_{i,j}x_j \le b_i$	$\sum_{j=1}^{n} a_{i,j}x_j + u_i = b_i, \quad u_i \ge 0$
Преобразование неравенства в равенство	$\sum_{j=1}^{n} a_{i,j}x_j \ge b_i$	$\sum_{j=1}^{n} a_{i,j}x_j - u_i = b_i, \quad u_i \ge 0$
Переход от решения задачи минимизации к решению задачи максимизации	$f = \sum_{j=1}^{n} c_j x_j \longrightarrow \min$	$-f = -\sum_{j=1}^{n} c_j x_j \longrightarrow \max$ $\min f = -\max(-f)$
Переход от свободной переменной к переменным, ограниченным по знаку	$x_j - \forall$	$x_j = x_j^1 - x_j^2 \ , x_j^1 \ge 0, x_j^2 \ge 0$

И еще одна задача линейного программирования

На складе фирмы имеется 50 ед. (погонных метров, квадратных футов или иных подходящих единиц измерения) дерева и 54 ед. (возможно, других) металла. Из этого сырья можно производить дачную мебель двух видов: столы и стулья. Пусть для изготовления одного стола требуется 10 ед. дерева и 6 ед. металла, а для изготовления одного стула — 5 ед. дерева и 9 ед. металла. Продажная цена одного стола составляет 8 условных денежных единиц, а одного стула — 6 у. е. Спрашивается, какое количество столов и стульев следует изготовить, чтобы максимизировать доход, уложившись при этом в имеющиеся ресурсы?

Представление задачи ЛП в каноническом виде

x1 - количество производимых столов;
X2 – количество производимых стульев;
Целевая функция:
$L(x_1, x_2) = 8x_1 + 6x_2 + x_3 + x_4$ —> max, =>
$L(x_1, x_2) = -8x_1 - 6x_2 - x_3 - x_4$ —> min

$10x_1 + 5x_2 + x_3 + x_4 = 50,$
$6x_1 + 9x + x_3 + x_4 = 54.$

$x_1>=0$, x2 >=0, x3 >=0, x4 >=0.

Представление задачи ЛП в матричном виде

x1 - количество производимых столов;
X2 – количество производимых стульев;
Целевая функция:
$L(x_1, x_2) = 8x_1 + 6x_2 + x_3 + x_4$ —> max, =>
$L(x_1, x_2) = -8x_1 - 6x_2 - x_3 - x_4$ —> min

10, 5, 0, 0
6,   9, 0, 0

$P_0=(50,54,0,0)^T$
$x_1>=0$, x2 >=0, x3 >=0, x4 >=0.

Основные понятия симплекс-метода

Опорное решение **вырожденное**, если в одной из базисных координат есть нуль

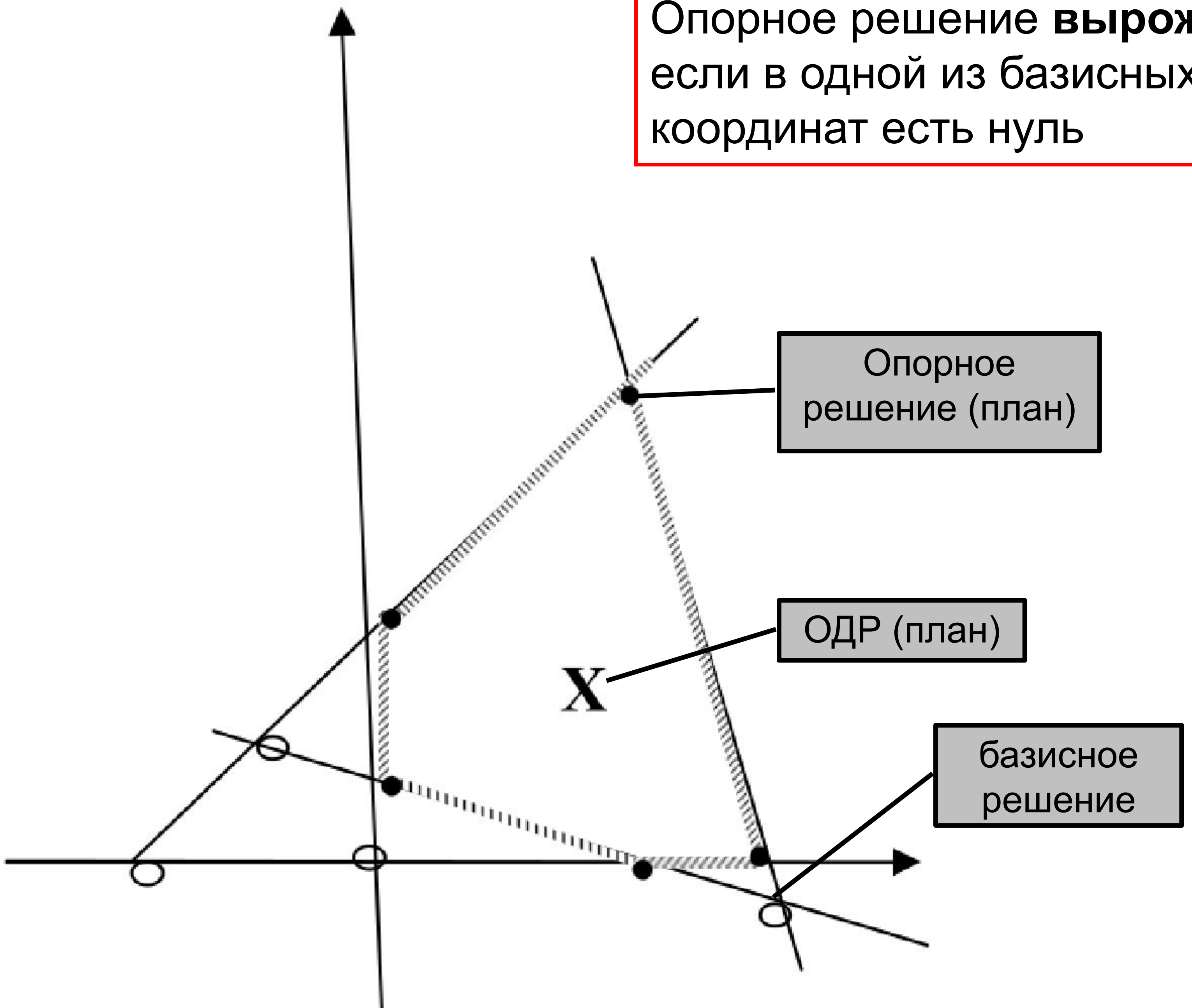


Схема симплекс-метода

1. Система уравнений записывается в канонической форме $\vec{X}_0 = (x_1, \ldots, x_m, 0, \ldots, 0)^T$
2. Строится исходных опорный план:
3. Проверка текущего плана на оптимальность

Все оставшиеся вектора разлагаются по базису, вычисляется целевая функция для них, выбирается для включения тот вектор, при котором целевая функция наименьшая

4. Вектор для исключения из базиса выбирается по формуле

$$\frac{x_l}{x_{lk}} = \min_{x_{ik}>0} \frac{x_i}{x_{ik}}, \quad i = 1, \ldots, m.$$

Пример решения задачи ЛП с помощью симплекс-метода (1)

Задача:

$$F = 2x_1 + 3x_2 - x_4 \to \max,$$

$$2x_1 - x_2 - 2x_4 + x_5 = 16,$$

$$3x_1 + 2x_2 + x_3 - 3x_4 = 18,$$

$$-x_1 + 3x_2 + 4x_4 + x_6 = 24,$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \ge 0.$$

Пример решения задачи ЛП с помощью симплекс-метода (2)

Задача уже приведена к каноническому виду. Выбираем в качестве переменных единичного базиса x_5, x_3, x_6 и составляем первую симплекс-таблицу для начального плана: X_0=(0,0,18,0,16,24). Составим симплекс-таблицу:

Базис	План	x1	x2	x3	x4	x5	x6
x5	16	2	-1	0	-2	1	0
x3	18	3	2	1	-3	0	0
x6	24	-1	3	0	4	0	1
F	0	-2	-3	0	1	0	0

В последней строке есть элементы <0, выбираем элемент с наименьшей оценкой (-3), потом решающий элемент по соотношению

$$\min\left\{-;\frac{18}{2};\frac{24}{3}\right\}=8$$

Пример решения задачи ЛП с помощью симплекс-метода (3)

Перейдем к новому базису с помощью последовательности элементарных операций преобразования:
Прибавляем к первой строке третью, умноженную на 1/3.
Вычитаем из второй строки третью, умноженную на -2/3.
Прибавляем к последней строке третью.
Делим третью строку на 3.
Получаем новую таблицу, вводя в базис x2 и выводя из базиса x6 .

Базис	План	x1	x2	x3	x4	x5	x6
x5	24	5/3	0	0	-2/3	1	1/3
x3	2	11/3	0	1	-17/3	0	-2/3
x2	8	-1/3	1	0	4/3	0	1/3
F	24	-3	0	0	5	0	1

В последней строке есть отрицательные элементы, поэтому делаем еще один шаг симплекс-преобразования

$$\min\left\{\frac{24}{5/3};\frac{2}{11/3};-\right\}=\frac{6}{11}$$

Пример решения задачи ЛП с помощью симплекс-метода (4)

Базис	План	x1	x2	x3	x4	x5	x6
x5	16	2	-1	0	-2	1	0
x3	18	3	2	1	-3	0	0
x6	24	-1	3	0	4	0	1
F	0	-2	-3	0	1	0	0

Таблица после 1-го шага преобразования

Базис	План	x1	x2	x3	x4	x5	x6
x5	24	5/3	0	0	-2/3	1	1/3
x3	2	11/3	0	1	-17/3	0	-2/3
x2	8	-1/3	1	0	4/3	0	1/3
F	24	-3	0	0	5	0	1

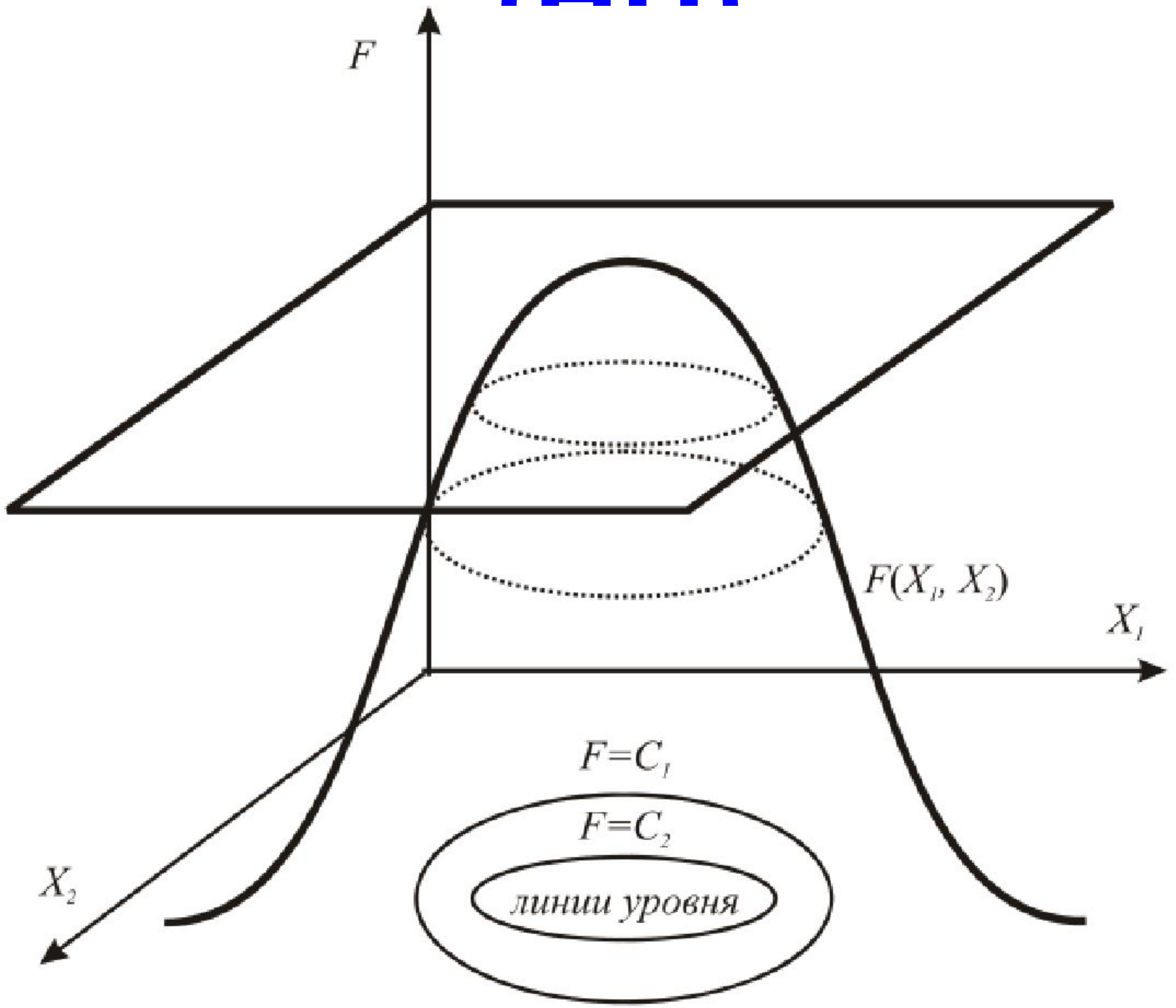
Пример решения задачи ЛП с помощью симплекс-метода (5)

Таблица после 2-го шага преобразования

Базис	План	x1	x2	x3	x4	x5	x6
x5	254/11	0	0	-5/11	21/11	1	7/11
x1	6/11	1	0	3/11	-17/11	0	-2/11
x2	90/11	0	1	1/11	9/11	0	3/11
F	282/11	0	0	9/11	4/11	0	5/11

В последней строке нет отрицательных элементов, значит, найдено оптимальное решение.
x_1=6/11, x_2=90/11, x_3=0, x_4=0, x_5=254/11, x_6=0, F_{max}=282/11

Нелинейное программирование (НП)



Постановка задачи нелинейного программирования

Необходимо минимизировать f(x)
при условиях:
$g_i(x) <= 0$, i=1..m,
$h_i(x) = 0$, i=1,..,l,
$x \in X$.
Здесь f, $g_1..g_m$, $h_1..h_l$ – функции, X – вектор компонент $x_1..x_n$.
$f(x)$ – целевая функция.
Необходимо минимизировать $f(x)$ при условии, что $x_1..x_n$ удовлетворяют заданным условиям. Т.е. необходимо найти такую точку $\bar{x}$, что $f(\bar{x}) \leq f(x)$.

Задачи НП в отличие от ЛП не имеют универсального метода решения !!!

Постановка задачи нелинейного программирования

Необходимо минимизировать (максимизировать) $f(x_1,x_2,..,x_n)$ при ограничениях:

$$g_1(x_1, x_2, \ldots, x_n) \geq 0;$$
$$g_2(x_1, x_2, \ldots, x_n) \geq 0;$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$g_m(x_1, x_2, \ldots, x_n) \geq 0;$$

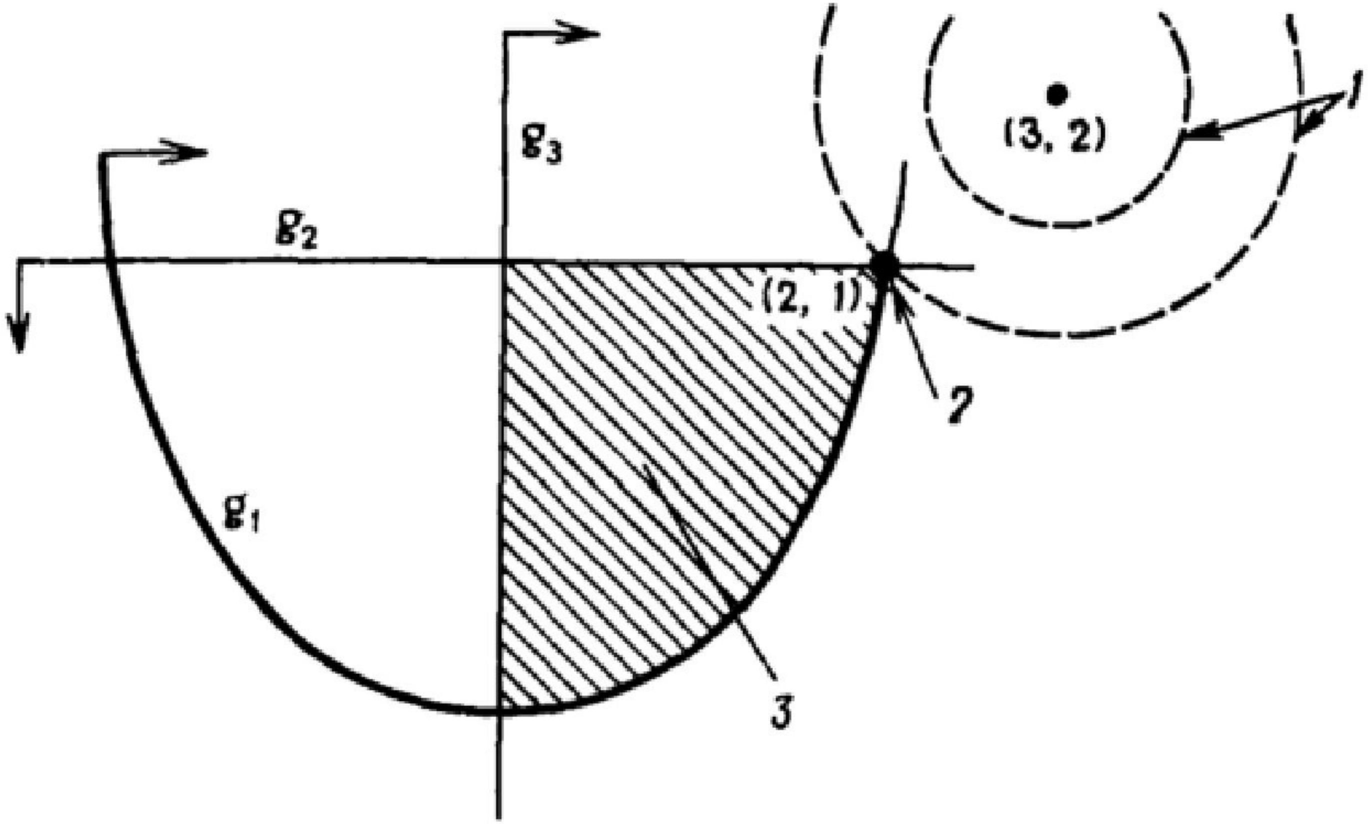
где функции

$$f(x_1, x_2, \ldots, x_n),\ g_i(x_1, x_2, \ldots, x_n) \geq 0,\ i = \overline{1, m}$$

нелинейны.

Пример задачи нелинейного программирования

Необходимо минимизировать функцию
$f(x)=(x_1-3)^2+(x_2-2)^2$,
при условиях:
$g_1(x)>=x_1^2-x_2-3$,
$g2(x)<=x_2-1$,
$g3(x)<=-x_1$.



Задача – найти
точку, находящуюся на минимальном расстоянии от точки (3,2) – центра круга $(x_1-3)^2+(x_2-2)^2=c$. радиус круга - $\sqrt{c}$. Такая точка – (2,1). Значение целевой функции в т. (2,1) равно 2.

Классификация методов нелинейного программирования

По количеству локальных критериев в целевой функции методы делятся на:

- однокритериальные,
- многокритериальные.

По длине вектора переменных делятся на:

- однопараметрические или одномерные (n=1),
- многопараметрические или многомерные (n>1).

По наличию ограничений:

- без ограничений (безусловная оптимизация),
- с ограничениями (условная оптимизация).

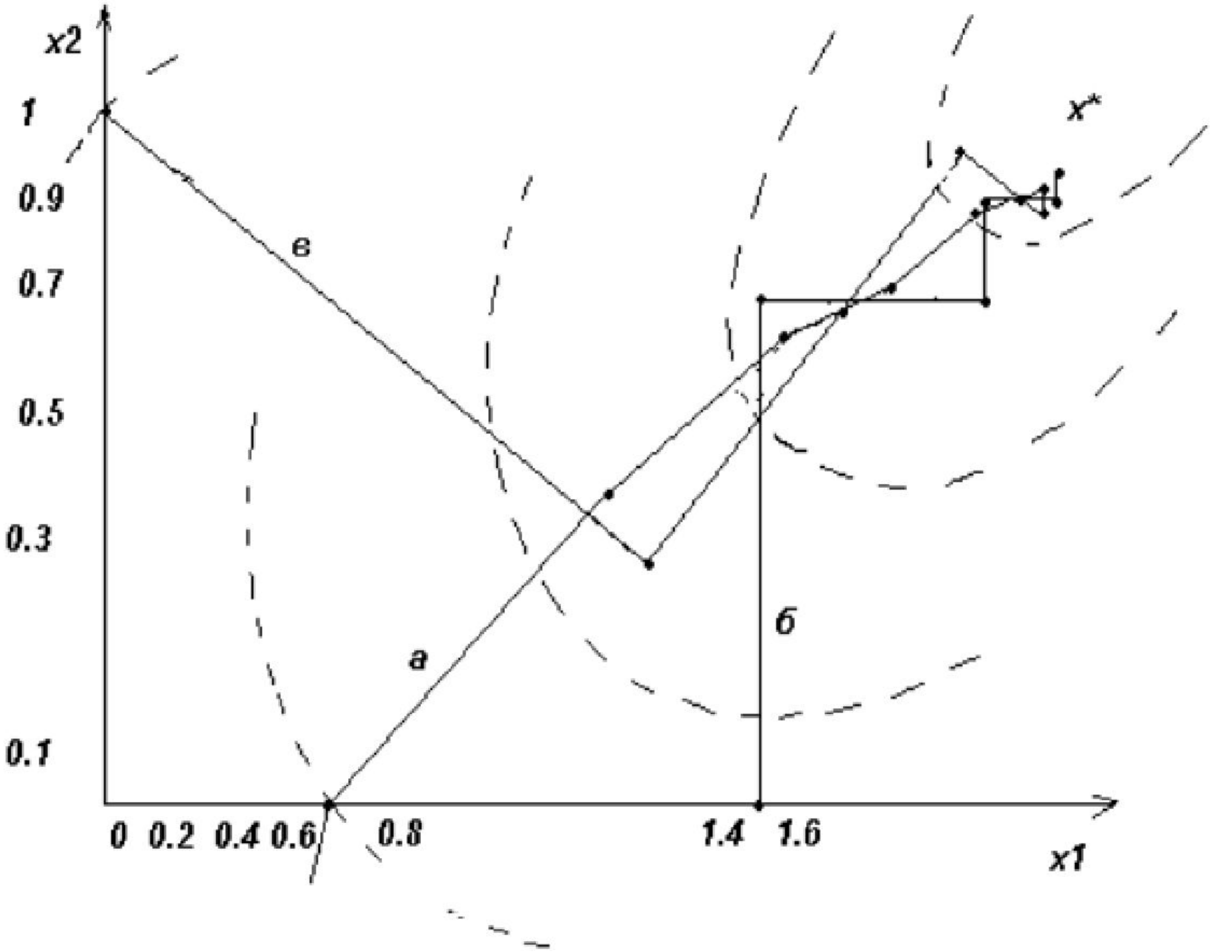
По типу информации, используемой в алгоритме поиска экстремума:

- методы прямого поиска, т.е. методы, в которых при поиске экстремума целевой функции используются только ее значения;
- *градиентные методы* первого порядка, в которых при поиске экстремума функции используются значения ее первых производных;
- *градиентные методы* второго порядка, в которых при поиске экстремума функции наряду с первыми производными используются и вторые производные.

Разделы в теории нелинейного программирования

- *выпуклое программирование* (исследование выпуклых и вогнутых функций),
- *квадратичное программирование* (линейные ограничения и квадратичная целевая функция),
- *целочисленное программирование* (неизвестные параметры могут принимать только целочисленные значения),
- *стохастическое программирование* (параметры принимают случайные значения, подчиняющиеся теории вероятности),
- *динамическое программирование* (время содержится как параметр) и др.

Градиентные методы нахождения оптимальной точки в функции нескольких переменных



а) – классический градиентный метод;
б) – покоординатного метод,
в) – метод наискорейшего спуска.

Метод циклического изменения переменных

Представляет собой процедуру рекурсивного перебора на множестве направлений поиска: каждый раз меняется только одна переменная. Затем вдоль каждого из координатных направлений последовательно проводится поиск точки экстремума на основе методов одномерной минимизации. Однако, если линии уровня ЦФ имеют овражный характер, то процедура поиска становится неэффективной и даже может привести к отсутствию сходимости к точке локального экстремума, если изменение координатных направлений поиска осуществляется в циклическом порядке (показано Пауэллом).

Классический градиентный метод

В качестве направления для изменения текущей точки выбирается вектор, направление которого противоположно направлению вектора градиента функции $\nabla f(x)$.

Вектор -*grad f* (X) = - $\nabla f(X)$ называется **антиградиентом** и является направлением наиболее быстрого ее убывания.

Рекурсивное соотношение для поиска новой точки будет:

$X_{k+1}=x_k-\lambda_k \nabla f(X)$,

где λ_k – величина шага на к-ой итерации.

k=1,2,..,

Метод наискорейшего спуска (метод Коши)

Если во время поиска шаг λ не меняется, то такой способ называется градиентным методом с дискретным шагом.

Процесс оптимизации на первых итерациях можно значительно ускорить, если λ_k выбирать из условия $\lambda_k = \min f(X_k + \lambda S_k)$, где для нахождения Sk используется любой метод одномерной оптимизации. Такой метод называется методом наискорейшего спуска.

Критерий остановки метода: $||\nabla f(X_k)|| < \varepsilon_k$

Недостаток метода – критерий остановки удовлетворяет не только точке экстремума, но и седловой точке.

Пример применения метода наискорейшего спуска(2)

Выполнение этого шага приведет в точку:

$$X_2 = \begin{pmatrix} -2 \\ 3 \end{pmatrix} - 0{,}164 \begin{pmatrix} -19 \\ 8 \end{pmatrix} = \begin{pmatrix} 1{,}116 \\ 1{,}688 \end{pmatrix}$$

Проверим критерий оптимальности: $|\nabla f(X_1)| = 2{,}02 > 0.1$

$\nabla f(X_2)$

Точность не достигнута, из точки делаем шаг вдоль направления антиградиента

$$X_3 = X_2 - \lambda_2 \nabla f(X_2) = \begin{pmatrix} 1{,}116 \\ 1{,}688 \end{pmatrix} - \lambda_1 \begin{pmatrix} 1{,}008 \\ 2{,}26 \end{pmatrix} = \begin{pmatrix} 1{,}116 - 1{,}008\lambda_1 \\ 1{,}688 - 2{,}26\lambda_1 \end{pmatrix}$$

$f(X_2) = 3(1{,}116 – 1{,}008\lambda_1)^2 + (1{,}688\text{-}2{,}26\lambda_1)^2 - (1{,}116 – 1{,}008\lambda_1)(1{,}688\text{-}2{,}26\lambda_1) - 4(1{,}116 – 1{,}008\lambda_1)$

$f'(X_2) = 11{,}76 – 6{,}12\lambda_1 = 0$

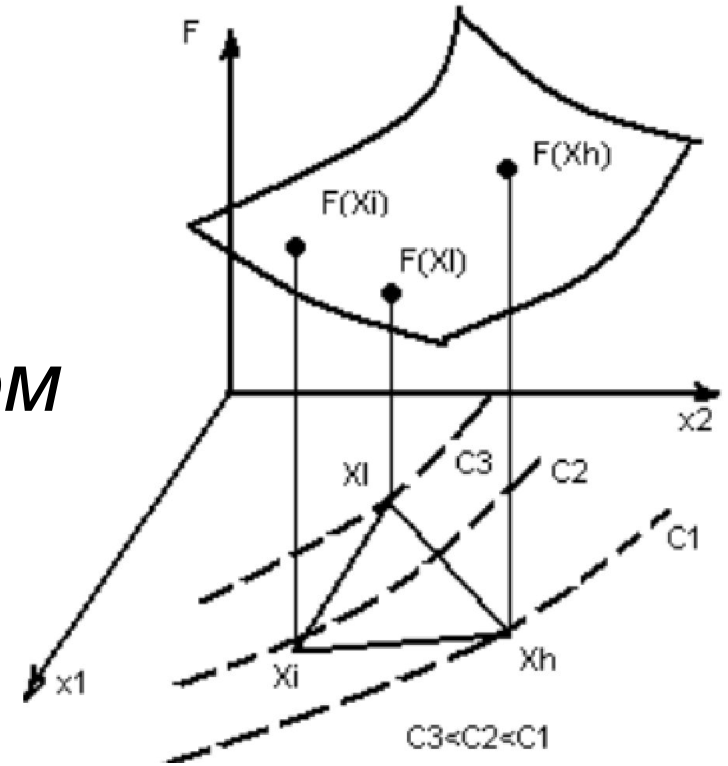
Получаем $\lambda_1 = 0.52$

$$X_3 = \begin{pmatrix} 1{,}116 \\ 1{,}688 \end{pmatrix} - 0{,}52 \begin{pmatrix} 1{,}008 \\ 2{,}26 \end{pmatrix} = \begin{pmatrix} 0{,}592 \\ 0{,}513 \end{pmatrix}$$

Метод деформируемого многогранника Нелдера-Мида

1) отражение - поворот симплекса через одну из сторон;
2) растяжение - если идём в правильном направлении;
3) редукция - возврат назад, если перескочили оптимум;
4) сжатие - уменьшение сторон, чтобы движение было с более мелким шагом.

Симплекс – многогранник в n-мерном пространстве



Динамическое программирование

Динамическое программирование (или динамическое планирование) представляет собой особый математический аппарат, позволяющий осуществлять оптимальное планирование управляемых процессов. Под «управляемыми» подразумеваются процессы, на ход которых мы можем в той или другой степени влиять. (Е.С. Вентцель)

История динамического программирования

Динамическое программирование возникло и сформировалось в 1950–1953 гг. благодаря работам Ричарда Беллмана и его сотрудников. Беллман (Bellman) Ричард Эрнест (1920-84) – американский математик. Первыми задачами, решаемыми с помощью динамического программирования были задачи управления запасами.

Задача динамического программирования

Задачи динамического программирования укладываются в следующую схему:

I. Имеется набор способов действий – допустимых управлений.

II. Имеется целевая функция («прибыль», «убыток») $S = S(u)$, где u пробегает допустимые управления.

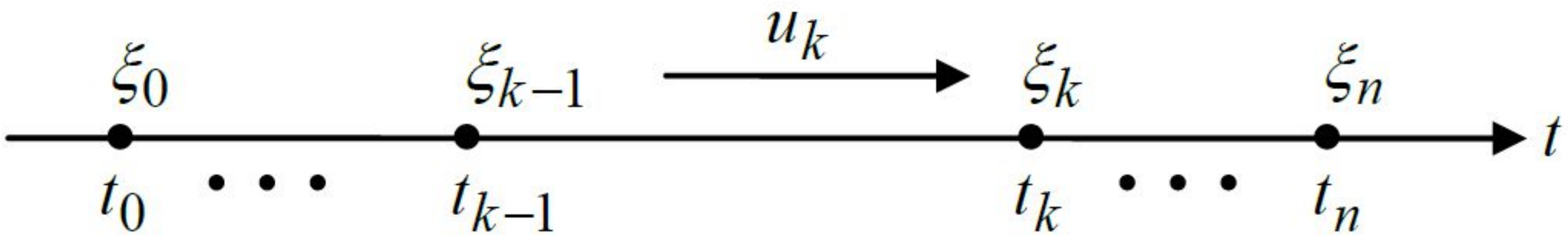
Требуется выбрать управление, которому отвечает оптимальное значение целевой функции.

Формализация задачи динамического программирования

Динамическая система (ДС) – такой объект, который может развиваться.
Квазидинамическая система – система, которую можно привести к динамической.
Пространство состояний (фазовое пространство) динамической системы – множество всех возможных состояний динамической системы.
ДС с дискретным временем – ДС, состояние которой меняется в дискретные моменты времени, например, t_0, t_1,t_2,..,t_n.
Траектория – набор состояний, через которые проходит ДС от начального к конечному состоянию.
Управляемая ДС – ДС, траекторию которой можно менять с помощью управляющих импульсов U_k.

Функционирование динамической модели

1. В начальный момент времени t_0 система находится в фиксированном состоянии ξ_0.
2. Переход $\xi_{k-1} \rightarrow \xi_k$ от состояния в момент t_{k-1} к состоянию в момент t_k (от t_0 к t_1 от t_1 к t_2 и т.д.) осуществляется под воздействие управляющих сигналов u_k, т.е. $\xi_k = \phi(\xi_{k-1}, u_k)$. Набор состояний ξ_0, ξ_1 ,…, ξ_n – траектория ДС. Причем, начало всех траекторий одинаковое - ξ_0.



Общая задача динамического программирования

Пусть имеется управляемая динамическая система оценивается каким-либо показателем, например, суммарным доходом $S=S(u_1,u_2,..,u_n)$. Суммарный доход равен сумме доходов на каждом шаге

$$S=f_1+f_2+\ldots+f_n. \quad (1)$$

f_k – доход на k-ом шаге, который зависит от состояния ДС в начале k-го шага и выбранного управления uk:

$$f_k= f_k(\xi_{k-1}\ , u_k) \quad (2)$$

Подставляя (1) в (2) получим:

$$S=\sum_{k=1}^{n} f_k(\xi_{k-1}, u_k)$$

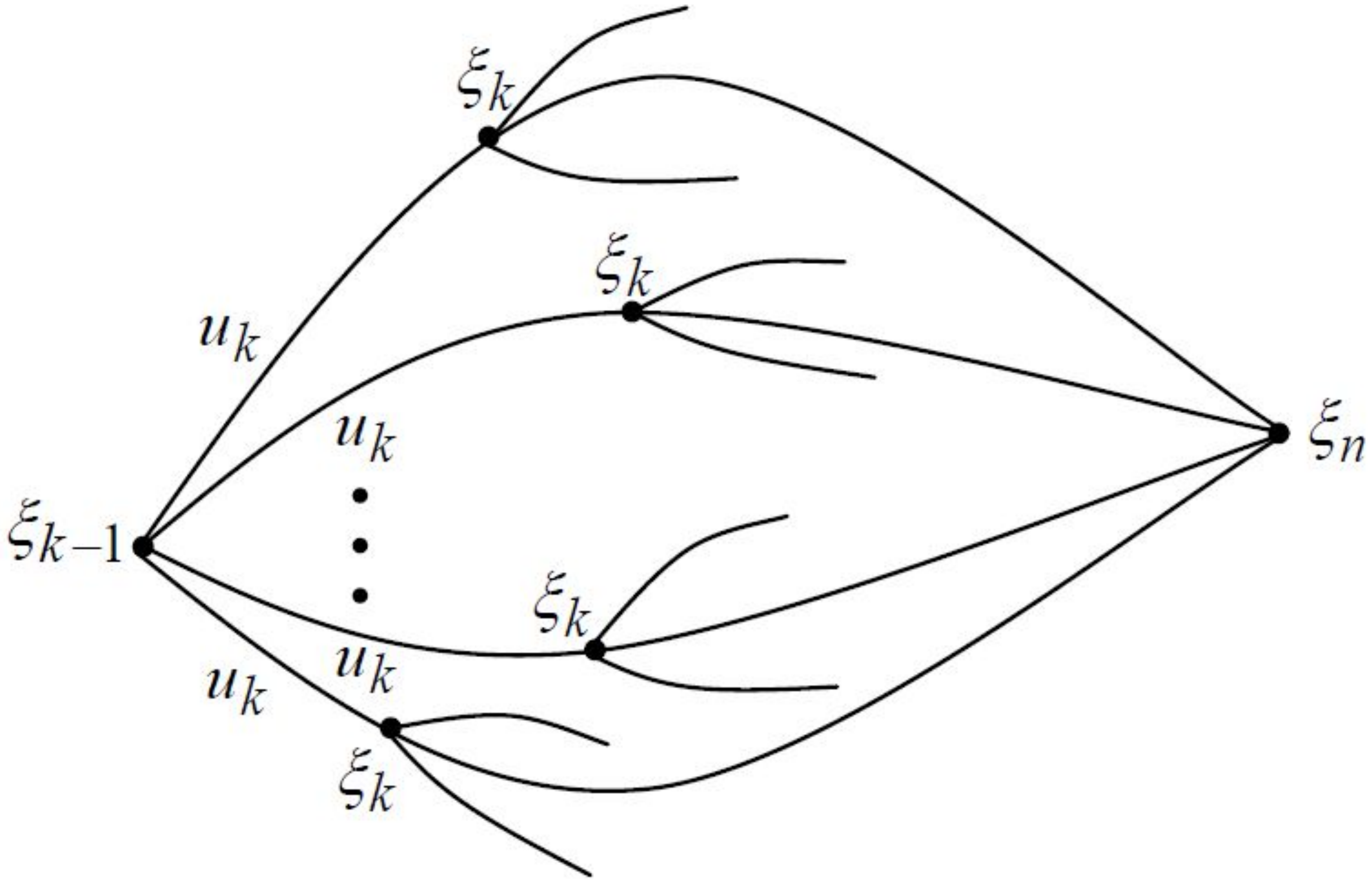
Такая функция называется **аддитивной целевой функцией**.

Общая задача динамического программирования

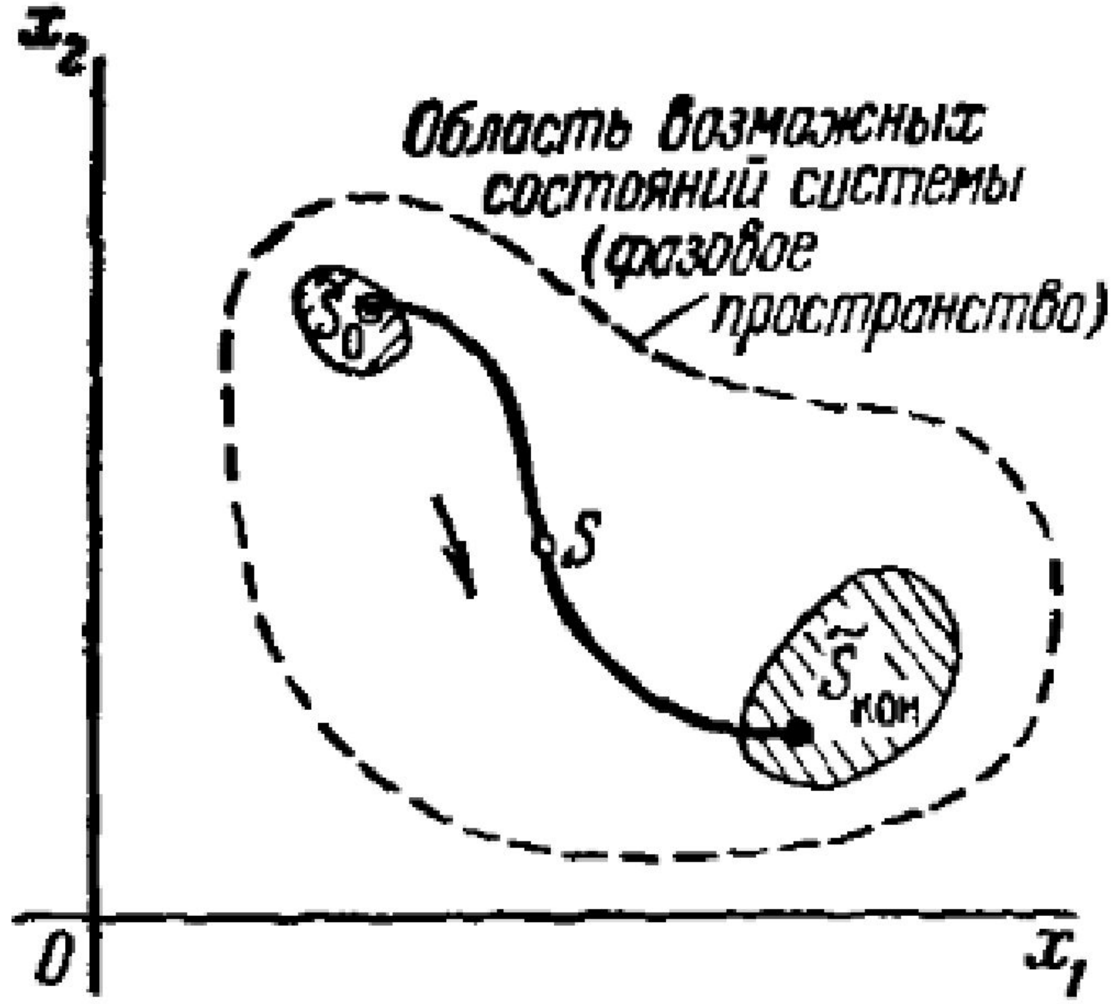
Для анализа имеется управляемая динамическая система, для которой выделено n шагов, а на каждом шаге выделены все возможные состояния (ξ_k), через которые проходит система, а также выделено начальное состояние ξ_0, а на каждом шаге заданы возможные управляющие воздействия (u_k). Также имеется аддитивная функция.

Задача ДП состоит в том, чтобы найти такую последовательность управляющих воздействий во время функционирования ДС ($u_1, u_2, \ldots, u_n$), чтобы аддитивная функция S достигала максимума или минимума. Траектория, на которой достигается max или min целевой функции, называется **оптимальной траекторией**. Задачей ДП является отыскание оптимальной траектории.

Метод решения задачи динамического программирования



Метод решения задачи динамического программирования



Метод решения задачи динамического программирования

Предположим, что к на- чалу k-го шага система оказалась в состоянии ξ_{k-1}. Оставшийся путь до конца система может проходить по различным траекториям в зависимости от выбора последующих управлений. Каждой траектории отвечает свой суммарный доход, т.е. доход на участке Обозначим этот доход через S_k.

$$S_k = f_k + f_{k+1} + \ldots + f_n$$

$S^*_k(\xi_{k-1})$ – **условный максимальный доход** – максимальный суммарный доход, начиная с k-го шага и до конца, т.е. на участке [k-1,n]. Тогда основную задачу ДП можно формал

$$S_k^*(\xi_{k-1}) = \max_{u_k}\{f_k(\xi_{k-1}, u_k) + S_{k+1}^*(\xi_k)\}$$

И требуется найти условное оптимальное управление на k-м шаге $u^*_k(\xi_{k-1})$.

Метод решения задачи динамического программирования

Задачи ДП решаются в два этапа:

1. движение от конца к началу

Начиная с кон- ца последовательно находим:

$$S_n^*(\xi_{n-1}), u_n^*(\xi_{n-1}); S_{n-1}^*(\xi_{n-2}), u_{n-1}^*(\xi_{n-2}); \ldots; S_1^*(\xi_0), u_1^*(\xi_0)$$

для всех возможных на соответствующих шагах состояний с использованием на каждом шаге, начиная с *n-1*-го, формулы для

$$S_k^*(\xi_{k-1}) = \max_{u_k}\{f_k(\xi_{k-1}, u_k) + S_{k+1}^*(\xi_k)\}$$

Метод решения задачи динамического программирования

Задачи ДП решаются в два этапа:

1. движение от конца к началу

Начиная с кон- ца последовательно находим:

$$S_n^*(\xi_{n-1}), u_n^*(\xi_{n-1}); S_{n-1}^*(\xi_{n-2}), u_{n-1}^*(\xi_{n-2}); \ldots; S_1^*(\xi_0), u_1^*(\xi_0)$$

для всех возможных на соответствующих шагах состояний с использованием на каждом шаге, начиная с *n-1*-го, формулы для

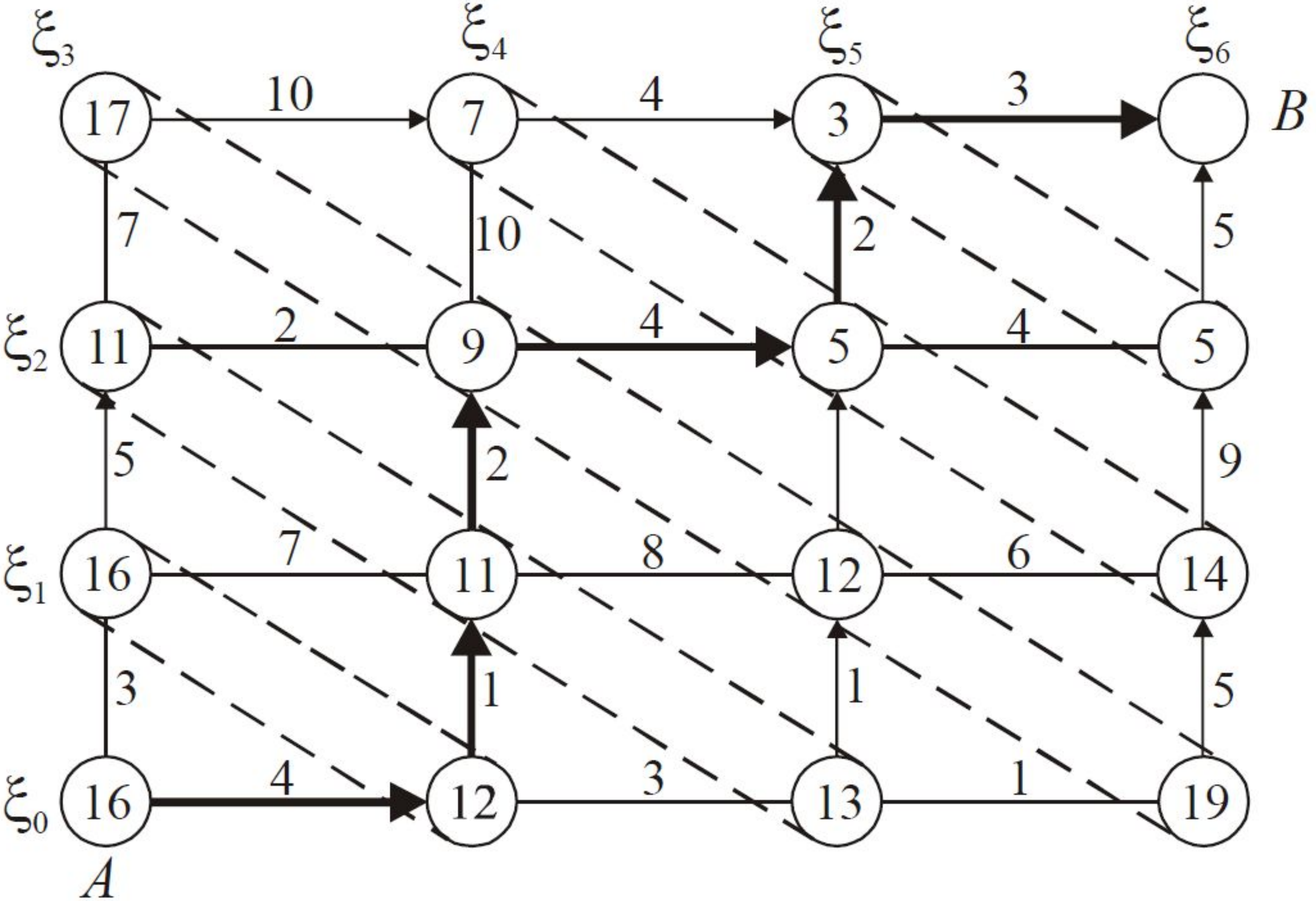
$$S_n^*(\xi_{n-1})$$

2 движение от начала к концу

Двигаясь от начала, существенно используя закрепленность начального состояния, строим безусловную оптимальную т

$$\xi_0 \xrightarrow{u_1^*(\xi_0)} \xi_1^* \xrightarrow{u_2^*(\xi_1^*)} \xi_2^* \longrightarrow \ldots \longrightarrow \xi_n^*$$

Задача об оптимальном маршруте



Задача об оптимальном маршруте

Требуется найти оптимальный маршрут на плоскости, проходящий через некоторые точки.

ДС представляет собой граф, где каждая вершина имеем четырех соседей. В нашем случае потребуется 6 шагов.

Состояние – узел графа. А – начальное состояние В – конечное состояние.

Под управлением будем понимать выбор направления движения: по горизонтали или вертикали.

Аддитивная целевая функция – величина потерь, приписанная к каждой дуге графа.

Целевая функция – суммарные потери на всех 6 шагах.

Решение:

Помечаем узлы графа последовательно от конца к началу минимальным весом, затем от начала к концу выбираем маршрут, с наименьшим весом для каждого шага.

Задача об оптимальной последовательности погрузки и разгрузки

Пусть на оптовую базу прибыло n машин с товаром для разгрузки и m машин для загрузки товаров, направляемых в магазины. Материально ответственное лицо оптовой базы осуществляет оформление документов по операциям разгрузки или загрузки одной машины, а затем переходит к обслуживанию другой машины. Издержки от операций обусловлены простоем транспорта, типом операции (прием или отгрузка товара). Необходимо спланировать последовательность операций обоих видов таким образом, чтобы суммарные издержки по приему и отправке товаров для всех машин были минимальными.

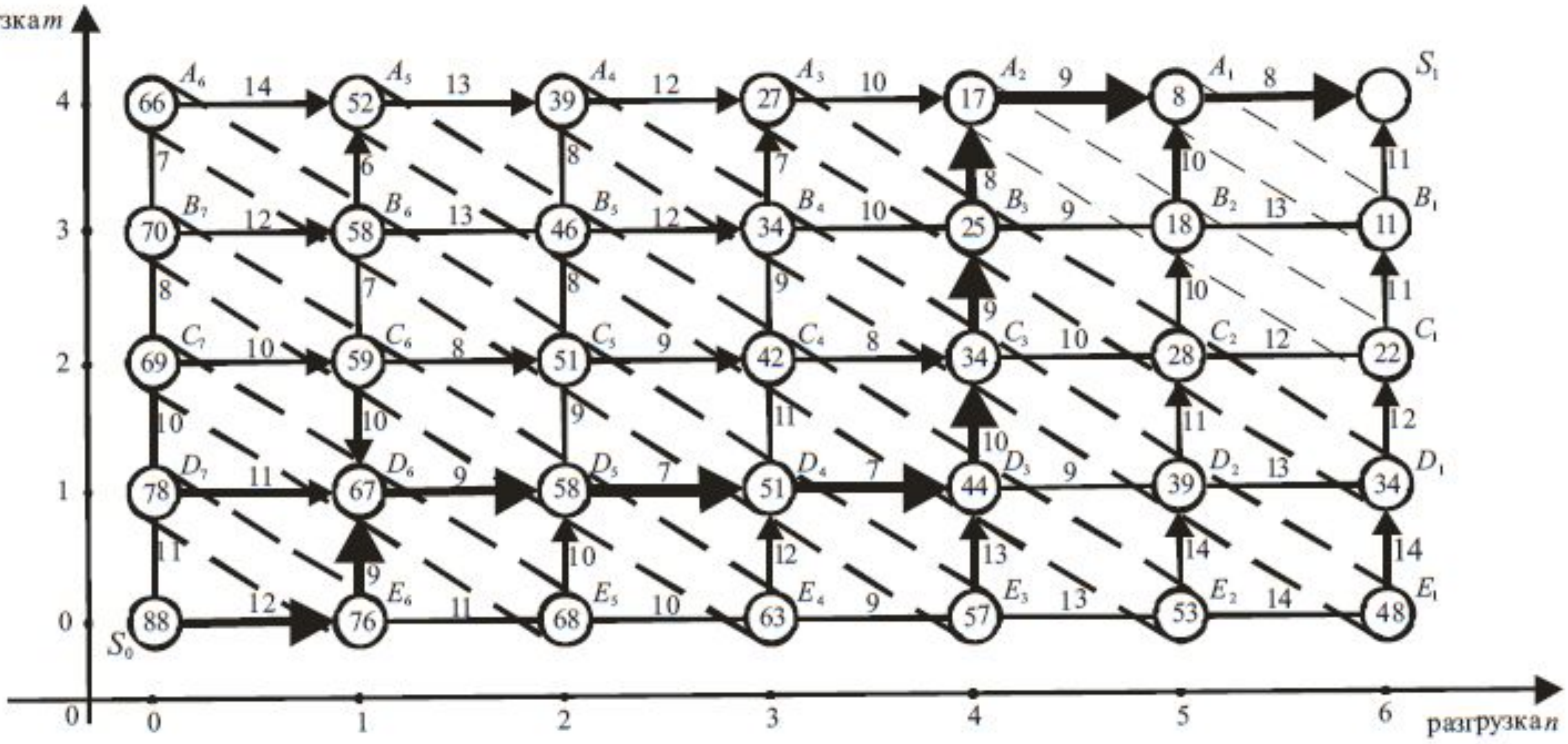
Задача об оптимальной последовательности погрузки и разгрузки

Если по оси X отложить число n разгруженных машин, а по оси Y число m загруженных товаром машин, то можно построить на плоскости граф состояний процесса, в котором каждая вершина характеризует состояние операции приема и отгрузки то- вара на оптовой базе. Ребра означают выполнение работы по приему или отправке товара на очередной машине. Каждому ребру можно сопоставить издержки, связанные с выполнением операции по разгрузке или загрузке машины.

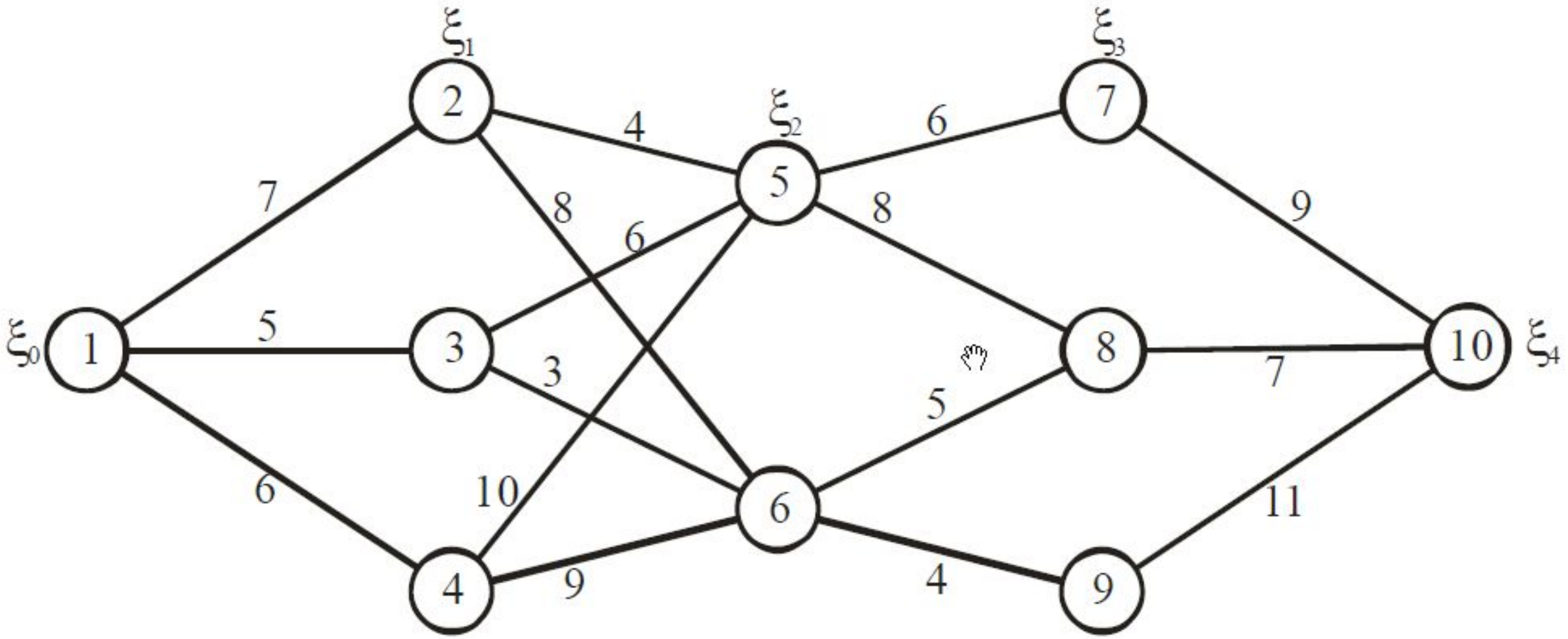
Таким образом, задачи сводится к ранее рассмотренной задаче выбора оптимального маршрута.

Задача об оптимальной последовательности погрузки и разгрузки

Пример. Пусть n=4, m=6 известны затраты по выполнению каждой операции, которые показаны на ребрах графа (рис. 4). Точка определяет начало процесса, точка конечное состояние, соответствующее приему и отправке всех

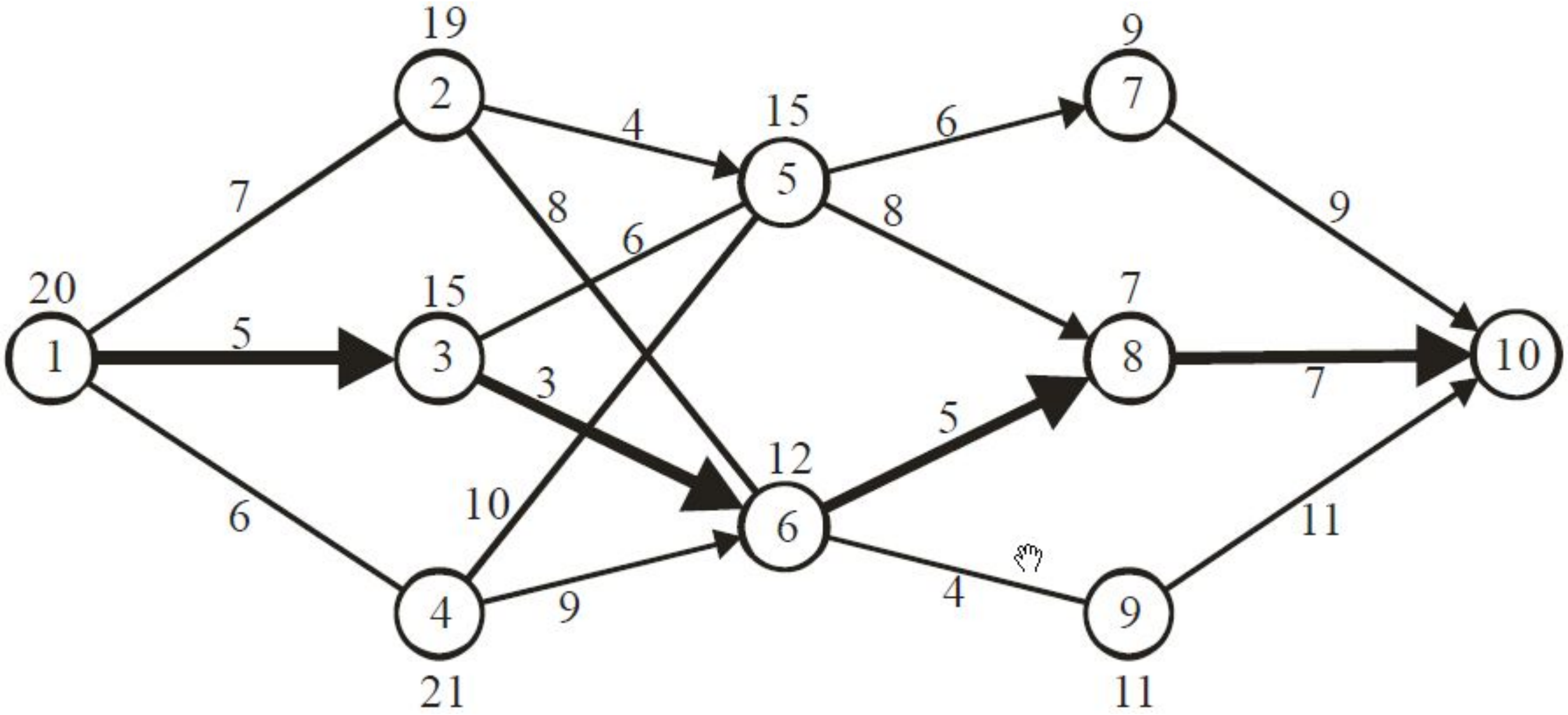


Задача о выборе оптимального пути на графе

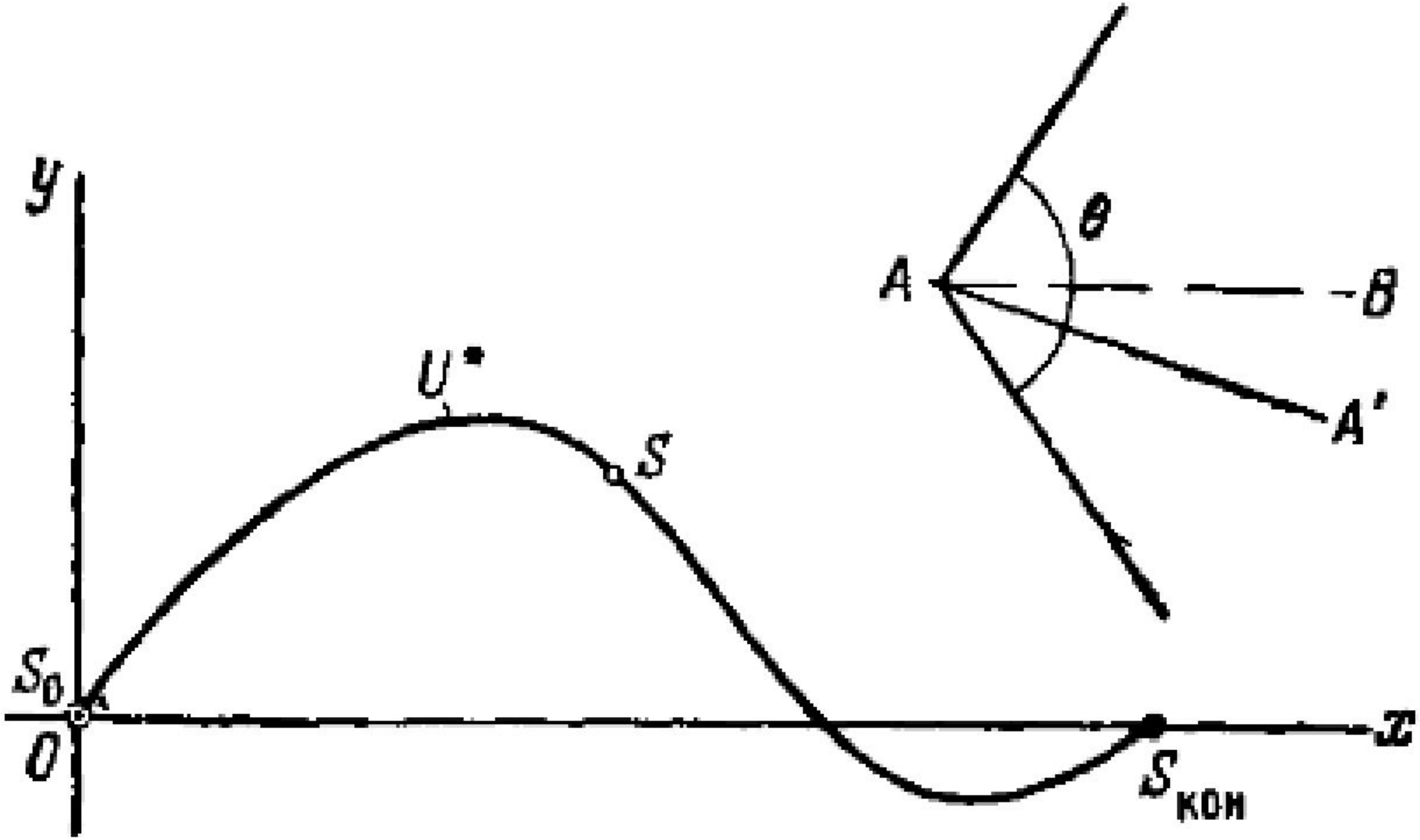


Необходимо выбрать путь из пункта 1 в пункт 10.
Двигаться можно только слева направо

Задача о выборе оптимального пути на графе



Задача прокладки непрерывного оптимального пути



Задача прокладки непрерывного оптимального пути

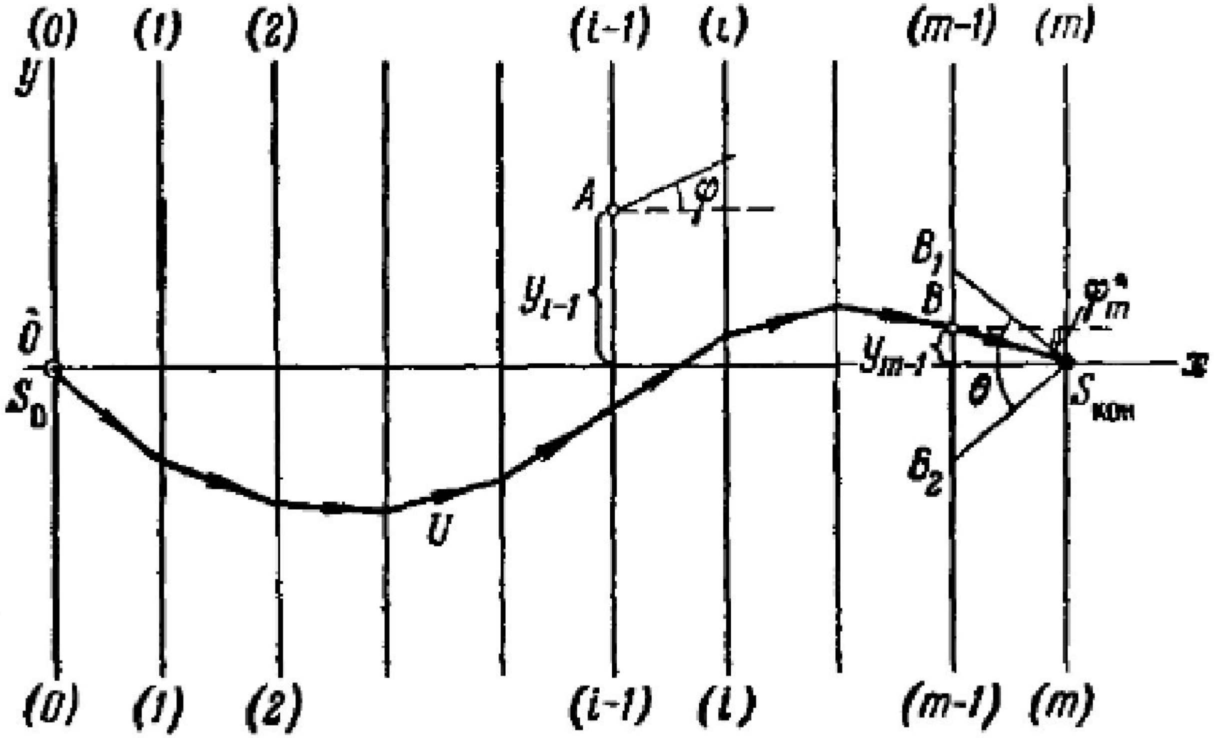
$$U_i^*(y_{i-1}) = \varphi_i^*(y_{i-1})$$

$$\varphi_m^* = \varphi_m^*(y_{m-1})$$

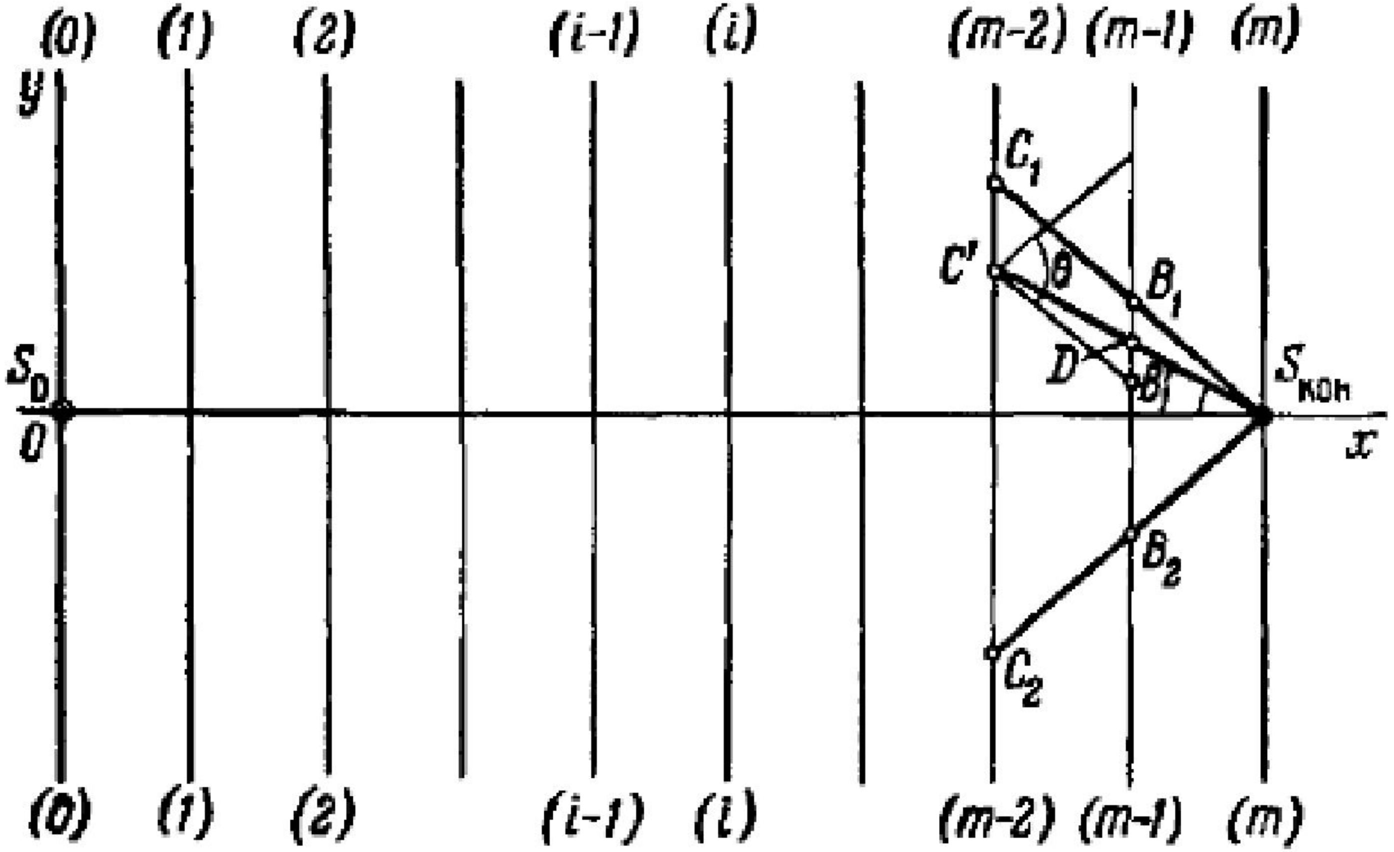
$$T_m^* = T_m^*(y_{m-1})$$

$$\varphi_m^*(y_{m-1}^{(1)}); \quad \varphi_m^*(y_{m-1}^{(2)}); \ \ldots$$

$$T_m^*(y_{m-1}^{(1)}); \quad T_m^*(y_{m-1}^{(2)}); \ \ldots$$

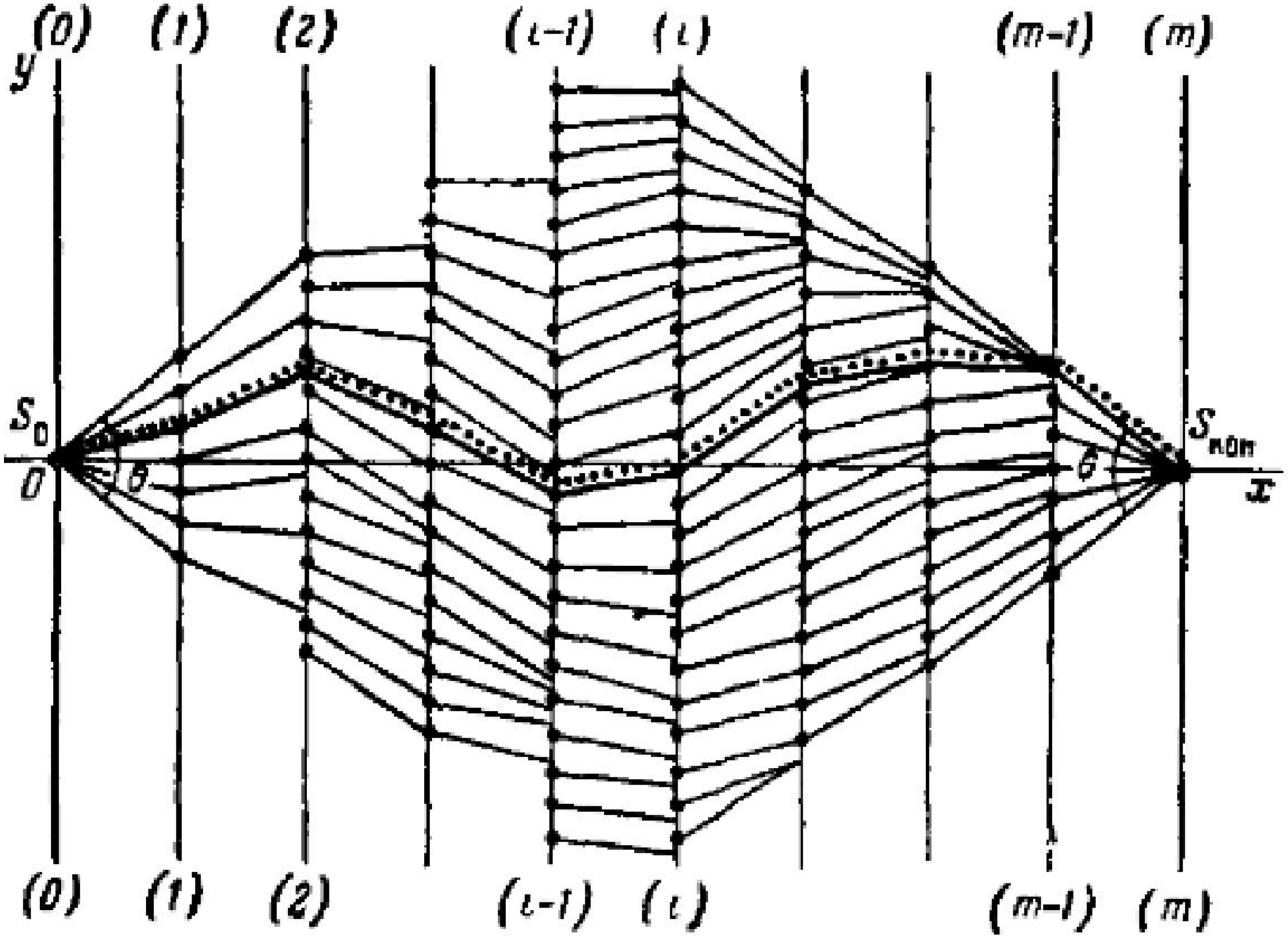


Задача прокладки непрерывного оптимального

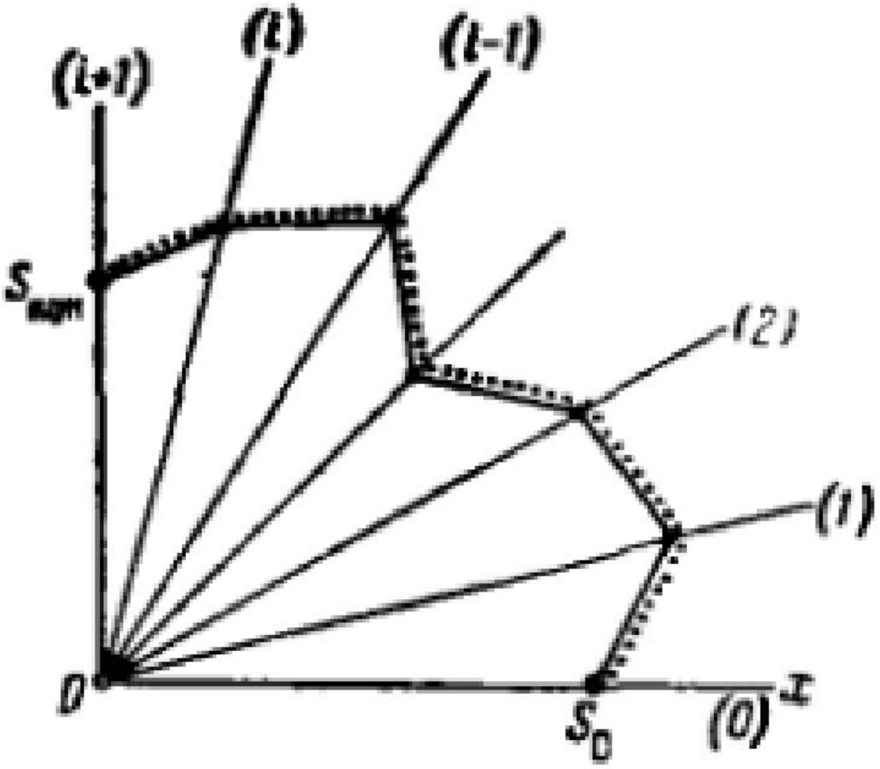


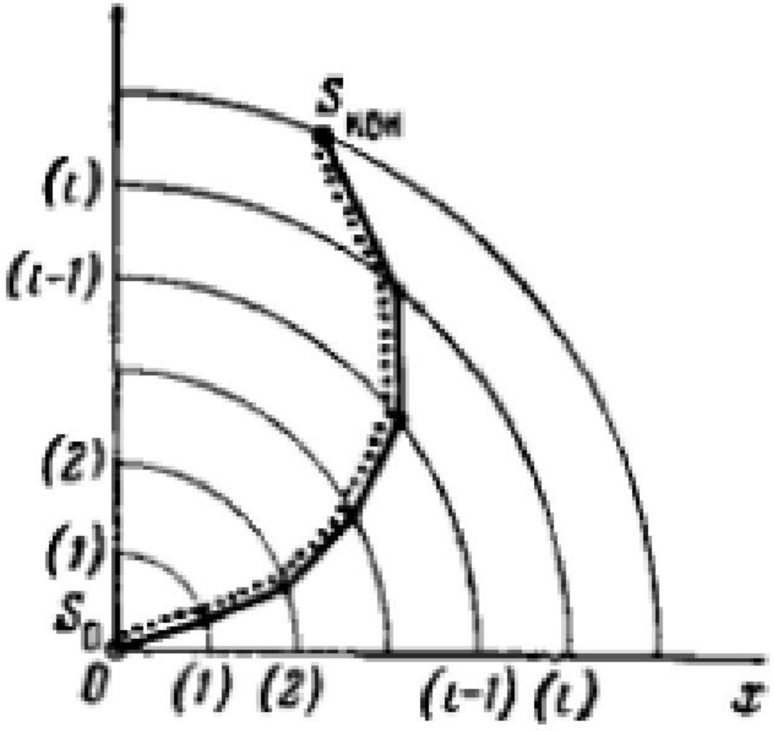
Зафиксируем на оси абсцисс несколько точек, в которых вычислим аддитивную целевую функцию.

Задача прокладки непрерывного оптимального пути

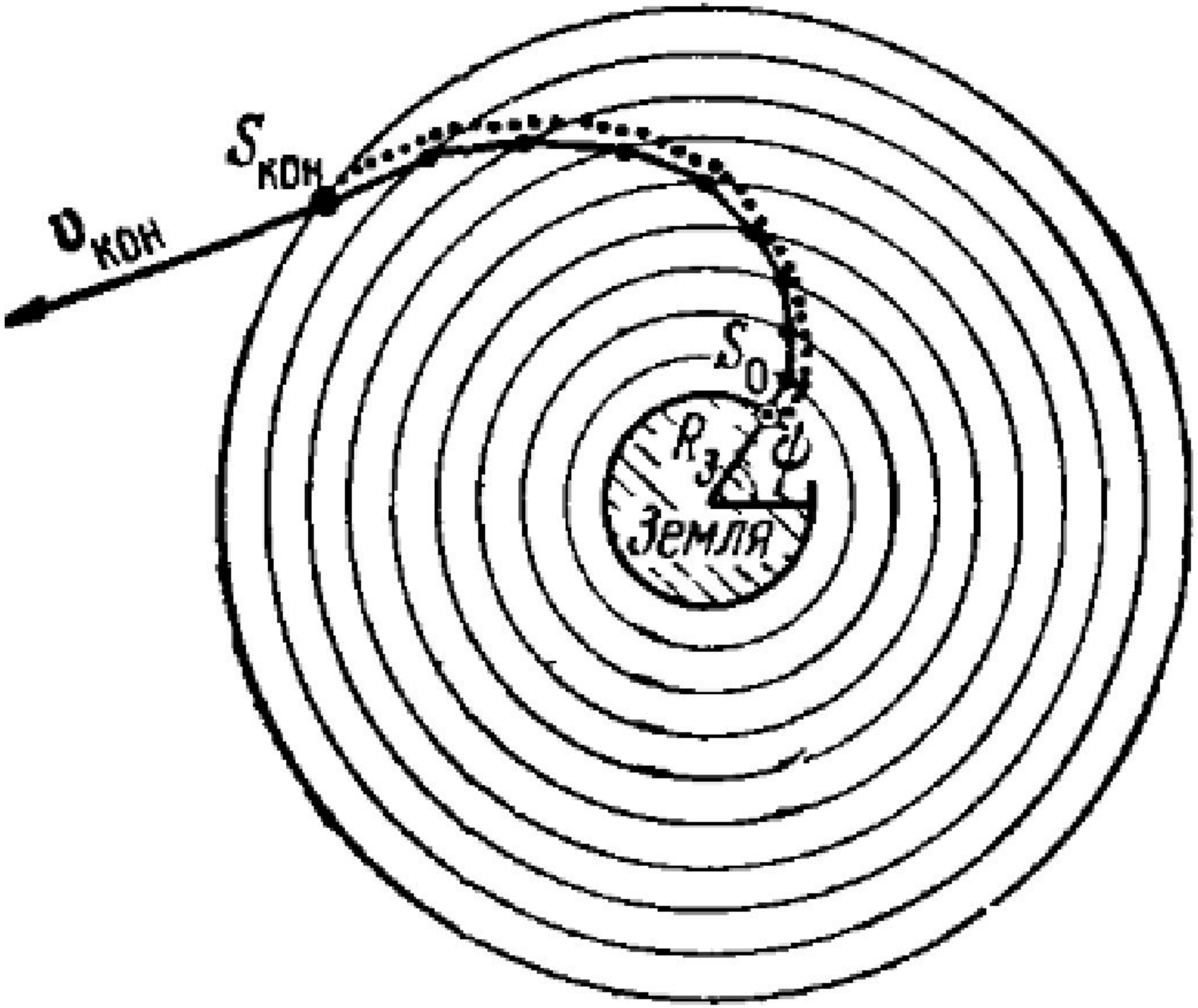


Задача прокладки непрерывного оптимального пути в полярной системе координат





Задача прокладки непрерывного оптимального пути в полярной системе координат



Задача о выборе оптимального оптимальной стратегии замены оборудования

Определить оптимальные сроки замены оборудования в течение n лет, при которых прибыль от эксплуатации оборудования максимальна, если известны: p – начальная стоимость оборудования; $R(t)$ – стоимость производимой продукции на оборудовании возраста t лет; $\mathrm{r}(t)$ – ежегодные затраты на эксплуатацию оборудования возраста t лет; $\phi(t)$ – ликвидная стоимость оборудования возраста t лет. Предполагаются, что к началу планового периода оборудование является новым.

Задача о выборе оптимального оптимальной стратегии замены оборудования

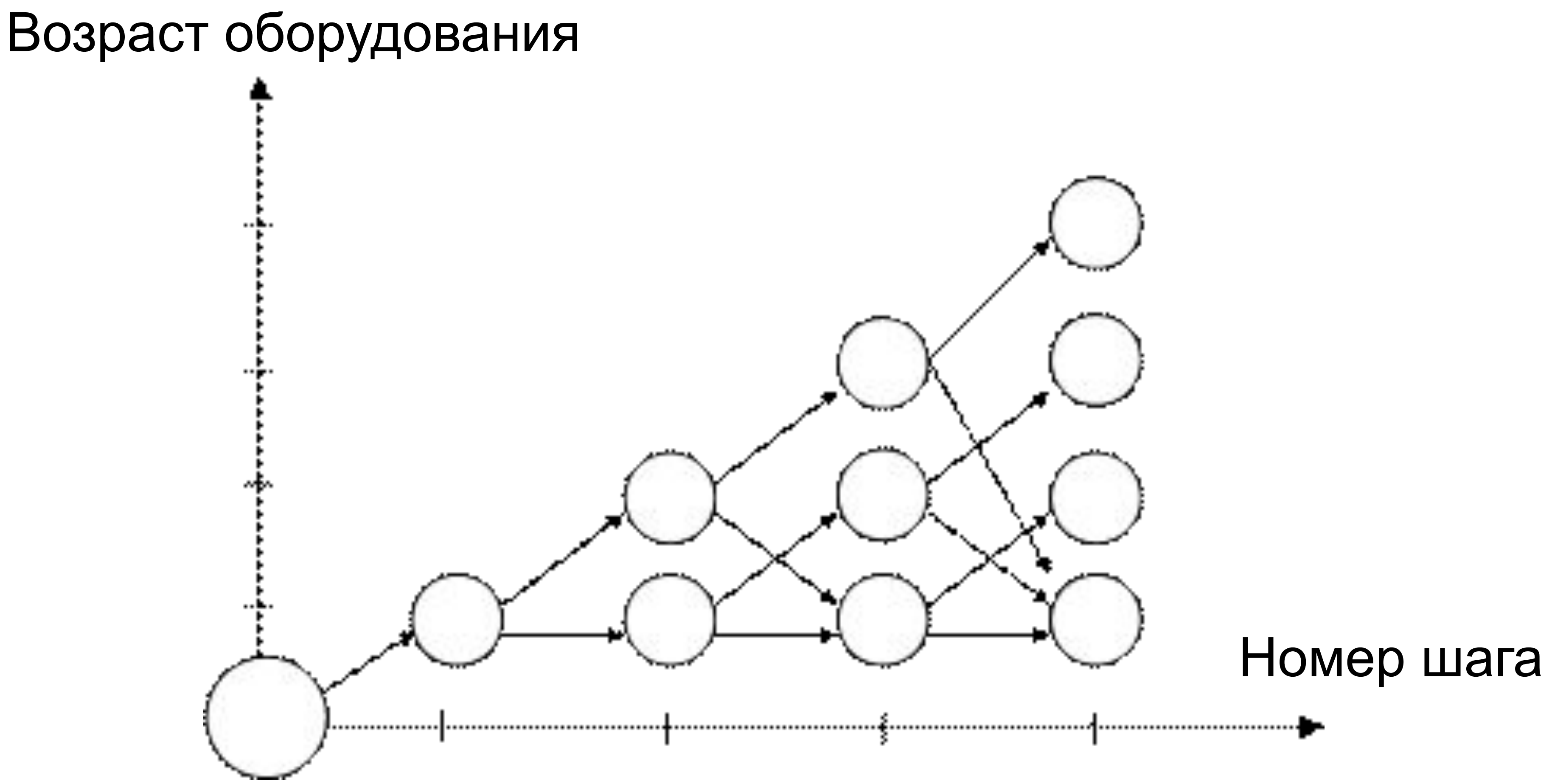
В качестве управления будем использовать решение о замене или незамене оборудования:

$$\xi_k = \begin{cases} \xi_{k-1} + 1, & u_k = u \\ 1, & u_k = \bar{u}. \end{cases}$$

$$f_k(\xi_{k-1}, u_k) = \begin{cases} R(\xi_{k-1}) - r(\xi_{k-1}) & при \quad u_k = u \\ R(0) - r(0) + \varphi(\xi_{k-1}) - p & при \quad u_k = \bar{u}. \end{cases}$$

$$S = \sum_{k=1}^{n} f_k(\xi_{k-1}, u_k)$$

Геометрическая интерпретация задачи о выборе оптимальной стратегии замены оборудования



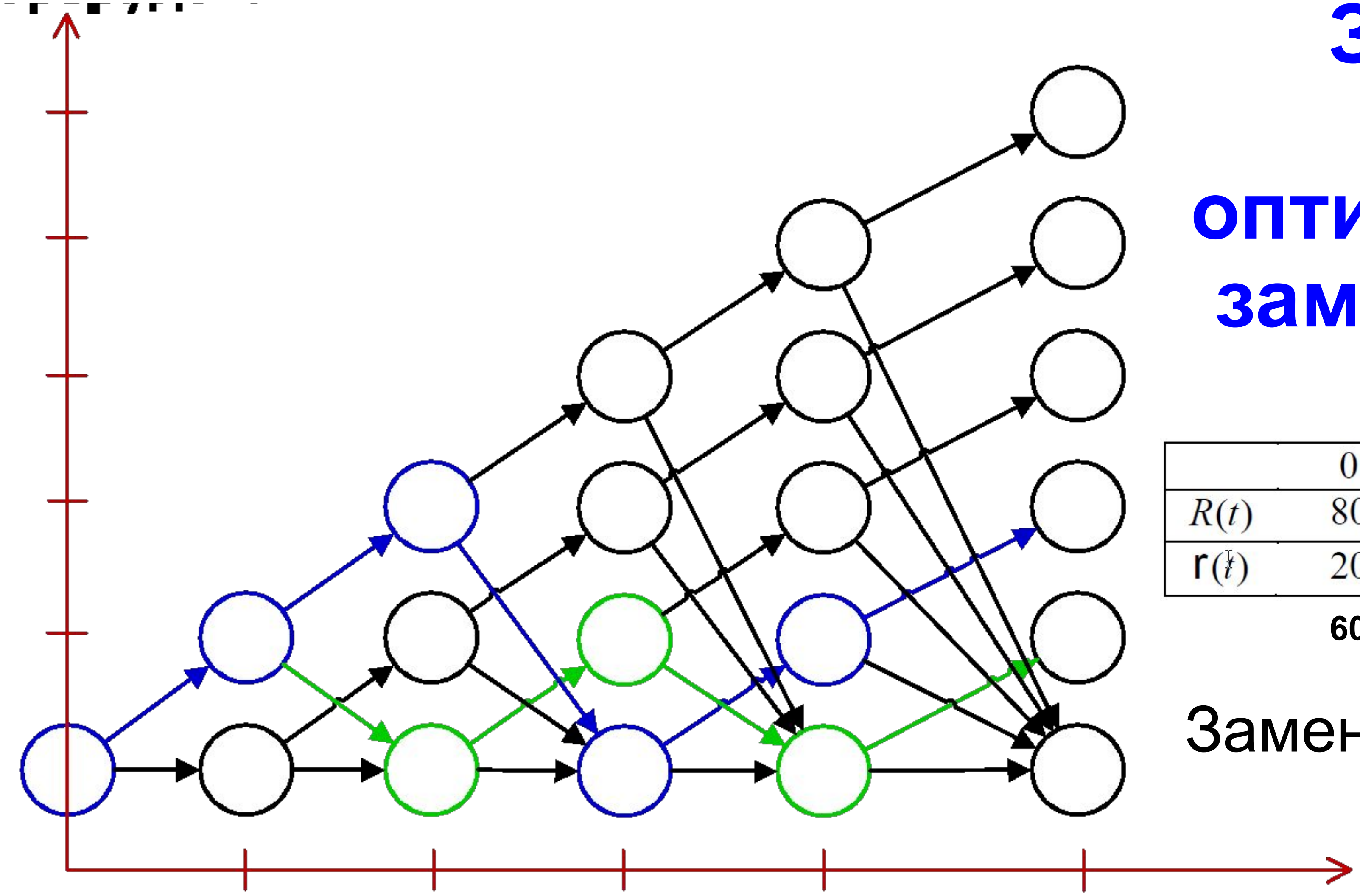
Задача о выборе оптимального оптимальной стратегии замены оборудования

	0	1	2	3	4	5
$R(t)$	80	75	65	60	60	55
r(t)	20	25	30	35	45	55

Замена оборудования p=40 у.е.

$$f_k(\xi_{k-1}, u_k) = \begin{cases} R(\xi_{k-1}) - \mathrm{r}(\xi_{k-1}) & \text{при} \quad u_k = u \\ 80 - 20 - 40 = 20 & \text{при} \quad u_k = \overline{u}. \end{cases}$$

Задача о выборе оптимального оптимальной стратегии замены оборудования



	0	1	2	3	4	5
$R(t)$	80	75	65	60	60	55
$r(t)$	20	25	30	35	45	55
	60	**50**	**35**	**25**	**15**	**0**

Замена оборудования p=40 у.е.

Оптимальные стратегии:

$\bar{u}$ u $\bar{u}$ u $\bar{u}$

$\bar{u}$ $\bar{u}$ u $\bar{u}$ $\bar{u}$

Задача распределения ресурсов

Объёмы	10	20	30	40	50
Подразделение 1	12	28	32	42	58
Подразделение 2	15	26	34	41	52
Подразделение 3	11	23	-	45	56
Подразделение 4	16	25	33	41	53

Необходимо вложить 80 у.е., чтобы получить максимальный доход

Выделяемые объёмы / Подразделения	10	20	30	40	50
1	12	28	32	42	58

1 / 2	10-12	20-28*	30-32	40-42	50-58
10–15*	20–27	30-43*	40-47	50-57	60–73*
20–26	30-38	40-54*	50-58	60-68	70–84*
30-34	40-46	50-62*	60-66	70-76	80–92*
40-41	50-53	60-69	70-73	80–83	
50-52	60-64	70-80	80-84		

Задача распределения ресурсов

3 \ 1+2	10-15	20-28*	30-43*	40-54*	50-62	60-73	70-78	80–92
10–11	20–26	30-39	40-54*	50-65	60–73	70–84	80–95	
20–23	30-38	40-51	50-66*	60-77*	70–85	80-96		
40-45	50-60	60-73	70-88*	80–99*				
50-56	60-71	70-84	80-99					

4 \ 1+2+3	30-43	40-54	50-66	60-77	70-88	80–99
10–16					80–104*	
20–25				80-102		
30-33			80-99			
40-41		80–95				
50-53	80-96					

Литература

1. Романовская, Адель Матвеевна Динамическое программирование: Учебное пособие. Романовская А.М., Мендзив М.В.– Омск: Издатель Омский институт (филиал) РГТЭУ, 2010. – 58 с.
2.Е.С. Вентцель Элементы динамического программирования Издательство «Наука», Москва, 1961