Список вопросов к экзаменационным билетам по дисциплине
"Методы оптимизации" - 2009 год.

1. Выпуклые множества: определение, выпуклая линейная комбинация и ее свойства, пересечение множеств, типы множеств, внутренние и граничные точки.
2. Выпуклые множества: крайняя точка, гиперплоскость, теорема о разделяющей гиперплоскости, опорная гиперплоскость, выпуклая оболочка.
3. Выпуклые функции: определения, свойство линейной формы, свойство суммы выпуклых функций, признак выпуклости дифференцируемой функции.
4. Выпуклые функции: свойство выпуклости области определения выпуклых функций, свойство глобальности минимума выпуклой функции.
5. Постановка задачи оптимизации. Классы оптимизационных задач: задачи безусловной оптимизации, условной оптимизации, классические на условный экстремум, выпуклые задачи оптимизации, задачи математического программирования.
6. Классы задач оптимизации: линейного программирования с примерами, квадратичного программирования, дискретного программирования, оптимального управления.
7. Условия экстремума одномерных функций без ограничений.
8. Условия экстремума многомерных функций без ограничений. Вид знакоопределенности квадратичной формы.
9. Классическая задача условной оптимизации, метод неопределенных множителей Лагранжа. (необходимые условия экстремума)
10. Геометрическая интерпретация множителей и метода Лагранжа, достаточные условия экстремума, седловые точки, решение задач с ограничениями - неравенствами классическим методом Лагранжа.
11. Понятие о численных методах оптимизации.
12. Одномерный пассивный поиск. Унимодальность, интервал неопределенности, принцип минимакса.
13. Принцип минимакса, расстановка экспериментов при пассивном поиске, метод дихотомии, эвристический алгоритм Свенна.
14. Метод Фибоначчи, метод золотого сечения.
15. Метод золотого сечения, методы оценивания с использованием квадратичной аппроксимации.
16. Метод средней точки, метод касательных, метод секущих.
17. Метод поиска по симплексу.
18. Метод поиска Хука-Дживса.
19. Метод сопряженных направлений Пауэлла.
20. Градиентные методы: с постоянным шагом, с дроблением шага.
21. Метод наискорейшего спуска, метод покоординатного спуска, сходимость градиентных методов.
22. Градиентный метод с масштабированием переменных.
23. Эвристические схемы градиентного метода.
24. Оптимизация многомерных функций методами второго порядка.
25. Метод сопряженных градиентов.
26. Теорема Куна-Таккера, доказательство достаточности.
27. Теорема Куна-Таккера, доказательство необходимости.
28. Развитие и обобщение метода Лагранжа, общая теорема математического программирования.
29. Общая теорема математического программирования, условия оптимальности для задач квадратичного программирования.
30. Метод Била.
31. Метод Вулфа.
32. Метод кусочно-линейной аппроксимации.
33. Метод проекции градиента.
34. Метод возможных направлений.
35. Методы штрафных функций.
36. Методы случайного поиска.
37. Постановка общей задачи линейного программирования, примеры задач.
38. Свойства решений задач линейного программирования.
39. Двойственные задачи линейного программирования и их свойства.
40. Идея метода последовательного улучшения плана, признак оптимальности.
41. Алгебраическое обоснование метода последовательного улучшения плана.
42. Метод искусственного базиса.
43. М-метод. Двойственный метод последовательного улучшения плана.
44. Понятие транспортной задачи по критерию стоимости и свойства таких задач.
45. Циклы в транспортной матрице. Связь между базисными и небазисными переменными в транспортной задаче.
46. Распределительный метод решения транспортной задачи. Методы получения первого допустимого базисного решения транспортной задачи.
47. Метод потенциалов для решения транспортной задачи в матричной постановке. Методы получения первого допустимого базисного решения для транспортной

задачи.
48. Усложненные постановки транспортных задач по критерию стоимости.Метод решения транспортных задач по критерию времени.
49. Основные понятия о графах и сетях.Метод решения задачи о кратчайшем пути.
50. Метод Форда-Фалкерсона для решения задачи о максимальном потоке в сети.
51. Линейная сетевая задача,метод потенциалов для ее решения.
52. Жордановы исключения.Геометрический метод решения задач линейного программирования.
53. Задачи оптимального управления.Принцип оптимальности динамического программирования.
54. Метод динамического программирования для дискретных систем.
55. Метод динамического программирования для непрерывных систем.
56. Решение задач распределения ресурсов методом динамического программирования.
57. Решение задачи о коммивояжере методом динамического программирования.

Перечень методов,включенных в задачи к экзаменационным билетам.

Условия экстремума одномерных и многомерных функций без ограничений.Метод Лагранжа,развитие метода Лагранжа,обобщение метода Лагранжа.Методы:золотого сечения,средней точки,Пауэлла,касательных,секущих. Методы поиска:по симплексу,Хука-Дживса.Градиентные методы:с постоянным шагом,наискорейшего спуска,Гаусса-Зейделя,Флетчера-Ривса. Методы второго порядка:
Ньютона и с регулировкой шага.
Методы решения линейных задач с ограничениями:
модифицированные Жордановы исключения,геометрический метод решения задач линейного программирования,метод последовательного улучшения плана,
метод искусственного базиса,М-метод,двойственный метод последовательного улучшения плана,распределительный метод решения транспортных задач,
метод потенциалов в матричной и сетевой постановках для решения транспортных задач,методы решения задач о максимальном потоке в сети и о кратчайшем пути.
Методы:Била,кусочно-линейной аппроксимации,Розена,внешних и внутренних штрафных функций,динамического программирования.

Вопрос 1.
Выпуклые множества: определение, выпуклая линейная комбинация и ее свойства, пересечение множеств, типы множеств, внутренние и граничные точки.

Рассмотрим точки n-мерного множества
$\mathrm{E}^n : \bar{x}^{(1)}, \bar{x}^{(2)} \ldots \bar{x}^{(n)}$ и составим линейную комбинацию этих векторов
$\mathrm{M}_i, i = 0,1,2 \ldots r$

$\begin{cases} \mathrm{M}_i \geq 0 \\ \sum^{r} \mathrm{M}_i = 1 \end{cases}$ -свойства вып. лин. комб. $\bar{x} = \sum_{i=1}^{r} \mathrm{M}_i * \bar{x}^{(i)}$ - выпуклая линейная комбинация

Такое множество называется выпуклым, если с двумя крайними точками оно содержит выпуклую комбинацию.

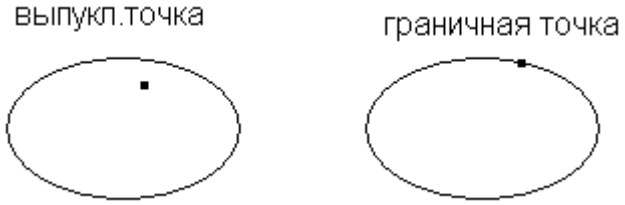
$G \subset E^{(n)} \qquad \underbrace{\bar{x}^{(1)}, \bar{x}^{(r)}}_{крайние_точки_отрезка}$

$\bar{x} = \grave{I} \quad * \bar{x}^{(2)} + (1-\mu) * \bar{x}^{(1)} \qquad 0 \leq \mu \leq 1$

$\bar{x} = (\mu_1 + \mu_2)(\frac{\mu_1}{\mu_1 + \mu_2} x^{(1)} + \frac{\mu_2}{\mu_1 + \mu_2} x^{(2)}) + \sum_{i=3}^{r} M_i x^{(i)}$ - выпуклая комбинация двух линейных векторов $y^{(1,2)}$

$\bar{x} = (\mu_1 + \mu_2)\bar{y}^{(1,2)} + \sum_{i=3}^{r} M_i x^{(i)}$

постепенно уменьшаем, в итоге: $\bar{x} = k_i * \bar{y}^{(1,r-1)} + \mu_r x_r \Rightarrow \quad \bar{x} \in G$ что и требовалось доказать



Вопрос 2.
Выпуклые множества: крайняя точка, гиперплоскость, теорема о разделяющей гиперплоскости, опорная гиперплоскость, выпуклая оболочка.

Точка $\bar{x} \in G \subset E^n$ называется крайней точкой множества, если не существует $\bar{x}^{(1)}, \bar{x}^{(2)} \in G$ чтобы для $\bar{x}$ выполнялось такое выражение:

$\bar{x} \neq Mx^{(2)} + (1-\mu)x^{(1)}, 0 < \mu < 1$

На прямой крайних точек не существует!!!

Гиперплоскость — гиперповерхность (в евклидовом n-мерном пространстве), которая задается одним линейным уравнением $(\bar{\lambda}; \bar{x}) = \sum_{j=1}^{n} \lambda_j x_j = c = const$

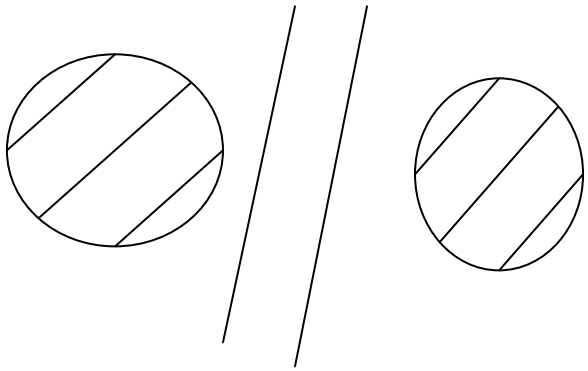
Произведение координат над гиперплоскостью:

$G_1 = (\lambda; x) \leq c \qquad G_2 = (\lambda; x) > c$

$\bar{x} \in G_1 : (\lambda; x) \leq c$

$\bar{x} \in G_2 : (\lambda; x) > c$

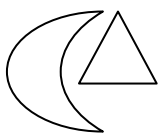
Теорема о разделяющей гиперплоскости.
Пусть G_1 и G_2 - замкнутые множества.



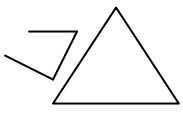
Случаи, когда теорема не верна:

1) выпукл.
2) замкнут.
3) без общих точек
4) не ограниченное

Посмотрим, что делается, если не выполнено одно из этих свойств:



не выпукл.



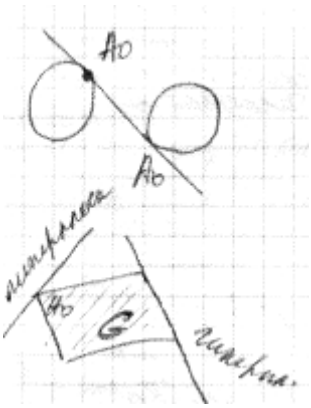
не замкнут.

$G_1 \bigcap G_2 = \varnothing$

G_1, G_2 - не огранич.

$(\bar{\lambda}; \bar{x}) = c$ - такая плоскость называется опорной в множестве G в точке λ_0.

1) $(\bar{\lambda}; \bar{A}_0) = c$
2) $(\bar{\lambda}; \bar{A}_0) \leq c$ или $(\bar{\lambda}; \bar{A}_0) \geq c$ при $\bar{x} \in G$



- должно находиться с одной стороны гиперплоскости

Выпуклой оболочкой множества $G \subset E^n$ называется такое множество:

$$[G]: \sum_{i=1}^{n} \mu_i x^{(i)}, \sum_{i=1}^{n} \mu_i = 1 \qquad \mu_i \geq 0 \quad x^{(i)} \in G$$

Свойства выпуклой оболочки:

1) Выпуклая оболочка – есть выпуклое множество

2) В выпуклую оболочку G входит выпуклое множество G: $G \subset [G]$

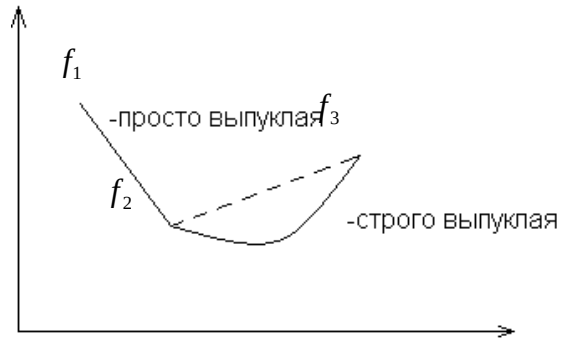
3) Если множество выпуклое, то выпуклая оболочка совпадает с множеством G

4) Если G - выпуклое, замкнутое, ограниченное, а G_1 – множество его крайних точек, то G является выпуклой оболочкой G_1

3. выпуклые функции: определения, свойства линейной формы, свойство суммы выпуклых функций, признак выпуклости дифференцируемой функции.

$f(\bar{x}) = f(x_1, x_2 x_3 ... x_n) \quad X \subset E^n$

Функция называется выпуклой, если для любых (·) $\bar{x}^{(1)}, \bar{x}^{(2)} \in X$ и любого числа $\lambda \in [0,1]$ выполняется неравенство $f\left[\lambda \bar{x}^{(2)} + (1-\lambda)x^{(1)}\right] \leq \lambda + (\bar{x}^{2}) + (1 - \lambda f(\bar{x}^{(1)}))$



Если неравенство выполняется наоборот, то функция вогнутая.
Если неравенство превращается в строгое равенство, то говорят что функция строго выпуклая (строго вогнутая).
Линейной формой называется линейная функция вида:

$f(\bar{x}) = \sum_{j=1}^{n} C_j x_j$ или $f(\bar{x}) = \bar{C}^T \bar{X}$ (транспонированная матрица);

$\bar{C} = (C_1, C_2 ..... C_n) \quad \bar{x} = (x_1, x_2 ..... x_n)$ - такая линейная комбинация называется линейной формой.

Свойства:

1. Линейная форма $f(\bar{x}) = \bar{C}^T \bar{X}$ является одновременно выпуклой и вогнутой на $E^{(n)}$

Доказательство:

Для любых 2 точек $\bar{x}^{(1)}, \bar{x}^{(2)} \in X$ выполняется неравенство:

$\bar{C}^T \left[\lambda \bar{x}^{(2)} + (1-\lambda)\bar{x}^{(1)}\right] \underset{\geq}{\leq} \lambda \bar{c}^T \bar{x}^{(2)} + (1-\lambda)\bar{c}^T \bar{x}^{(1)}$, что и указывает на принадлежность f(x) к классу выпуклых функций, но не к классу строго выпуклых функций.

2. Если несколько функций $f_j(\bar{x})$ выпуклые на некотором выпуклом множестве, то

$f(\bar{x}) = \sum_{j=1}^{m} f_j(\bar{x})$ тоже является выпуклой функцией.

Доказательство:

Для любых (·) $\bar{x}^{(1)}, \bar{x}^{(2)} \in X$ и любого числа $\lambda \in [0,1]$ выполняется неравенство

$$f\left[\lambda \bar{x}^{(2)} + (1-\lambda)x^{(1)}\right] = \sum_{j=1}^{m} f_i \left[\lambda \bar{x}^{(2)} + (1-\lambda)x^{(1)}\right] \leq \sum_{j=1}^{m} \left[\lambda f_i \bar{x}^{(2)} + (1-\lambda) f_i x^{(1)}\right] = \lambda \sum_{j=1}^{m} f_i \bar{x}^{(2)} + (1-\lambda) \sum_{j=1}^{m} f_i x^{(1)}$$

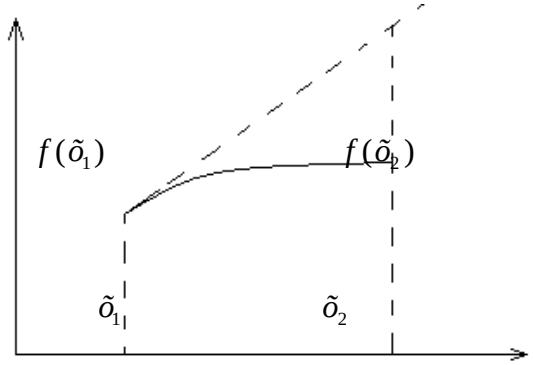
$$= \lambda f \bar{x}^{(2)} + (1-\lambda) f x^{(1)}$$

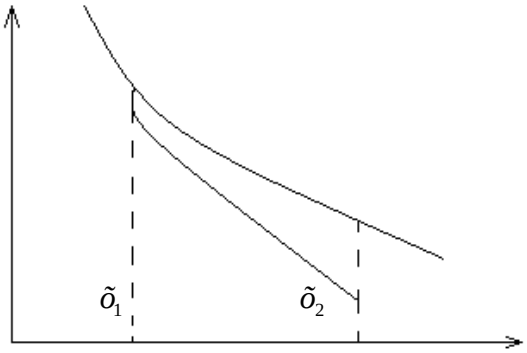
что и требовалось доказать.

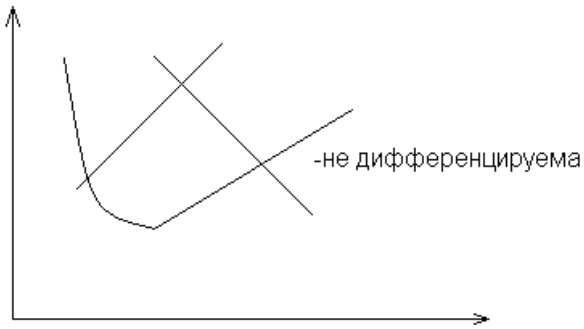
3. Функция дифференцируемая $f^{|}(\bar{x})$ (существует производная) , то справедливо неравенство: $f(\bar{x}^{(2)}) - f(\bar{x}^{(1)}) \geq f^{|}(\bar{x}^{(1)}) * (\bar{x}^{(2)} - \bar{x}^{(1)})$, где $f^{|}(\bar{x}^{(1)})$ - вектор градиент.

Доказательство:

Из формулы $f\left[\lambda\bar{x}^{(2)}+(1-\lambda)x^{(1)}\right]\le\lambda+(\bar{x}^{2})+(1-\lambda f(\bar{x}^{(1)}))$ следует что при $\lambda\in[0,1]$ $\frac{f\left[\bar{x}^{(1)}+\lambda(\bar{x}^{(2)}-\bar{x}^{(1)})\right]-f(\bar{x}^{(1)})}{\lambda}\le f(\bar{x}^{(2)})-f(\bar{x}^{(1)})$ если разложить числитель этой дроби по формуле Тейлора без учета остаточных членов, получим:
$f\left[\bar{x}^{(2)}+\lambda(x^{(2)}-x^{(1)})\right]=f(x^{(1)})+\lambda f^{|}\left[x^{(1)}+\lambda\theta(x^{(2)}-x^{(1)})\right]*(x^{(2)}-x^{(1)})$ где $\theta\in[0,1]$ и при $\lambda\to\infty$ окончательно получим: $f^{|}(\bar{x}^{(0)})(\bar{x}^{(2)}-\bar{x}^{(1)})\le f(\bar{x}^{(2)})-f(\bar{x}^{(1)})$ где $\bar{x}^{(1)}$ - любая внутренняя точка из множества X.







4. Выпуклые функции: свойства выпуклости области определения выпуклых функций, свойство глобальности минимума выпуклой функции.

Теорема:

$g_i(x)$ - выпуклая функция и $\bar{R}$ - множество точек, удовлетворяющих условиям:

$\bar{x} \in \bar{R}; __ g_i(\bar{x}) \le b_i; __ \bar{x} \ge 0;$

То $\bar{R}$ - выпуклое множество;

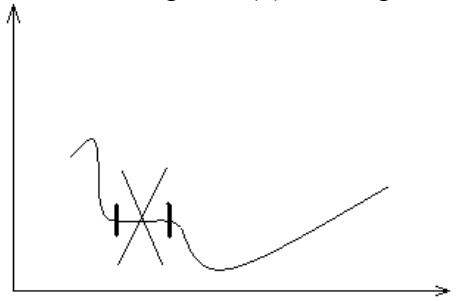
Доказательство:

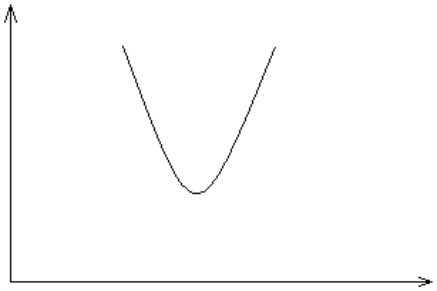
Пусть $\bar{x}^{(1)}, \bar{x}^{(2)} \in \bar{R}$. Рассмотрим точку $\tilde{x}$, представляющую собой линейную выпуклую комбинацию $\bar{x}^{(1)}, \bar{x}^{(2)}$. $\tilde{x} = \lambda \bar{x}^{(2)} + (1-\lambda)\bar{x}^{(1)}$. Теорема будет доказана в том случае, если удастся показать что $\tilde{x}$ принадлежит $\bar{R}$, а в этом случае $\bar{R}$ выпукло, т.к. вместе с любыми двумя своими точками содержит и выпуклую линейную комбинацию этих точек.. очевидно, что $\tilde{x} \ge 0$, осталось показать, что $g_i(\tilde{x}) \le b_i$, а это видно из выпуклости $g_i(x)$. Действительно: $g_i(\tilde{x}) = g_i\left[\lambda \bar{x}^{(2)} + (1-\lambda)\bar{x}^{(1)}\right] \le \lambda g_i \bar{x}^{(2)} + (1-\lambda) g_i \bar{x}^{(1)}$. Из условия $\bar{x} \in \bar{R}; __ g_i(\bar{x}) \le b_i; __ \bar{x} \ge 0$ следует , что при $\lambda \in [0,1]$ справедливы условия $\lambda g_i(\bar{x}^{(2)}) \le \lambda b_i$ $(1-\lambda) g_i(\bar{x}^{(1)}) \le (1-\lambda) b_i$. Следовательно: $\lambda g_i(\bar{x}^{(2)}) + (1-\lambda) g_i(\bar{x}^{(1)}) \le \lambda b_i (1-\lambda) b_i = b_i$, то есть $g_i(\tilde{x}) \le b_i$, следовательно $\bar{R}$ - внутреннее множество.

Теорема :

Если f(x) – выпуклая функция, заданная на выпуклом замкнутом множестве, тогда любой относительный (локальный, местный) минимум является абсолютным минимумом f(x) на $X \subset E^n$.

Доказательство: ведем от противного. Предположим, что наряду с относительным минимумом в точке $\bar{x}^{(0)}$ функция имеет абсолютный минимум в точке $\bar{x}^*$, так что $f(\bar{x}^{(0)}) > f(\bar{x}^*)$. Из условия выпуклости функции следует что для любого $\lambda \in [0,1]$ выполняется неравенство: $f\left[\lambda \bar{x}^* + (1-\lambda) x^{(0)}\right] \le \lambda f(\bar{x}^*) + (1-\lambda) f(\bar{x}^{(0)}))$, если учесть, что $f(\bar{x}^{(0)}) > f(\bar{x}^*)$, то можно записать : $f\left[\lambda \bar{x}^* + (1-\lambda) x^{(0)}\right] = \lambda f(\bar{x}^*) + (1-\lambda) f(\bar{x}^{(0)}) = f(\bar{x}^{(0)})$. Рассмотрим ξ окрестность точки $\bar{x}^{(0)}$ с $\xi < \| \bar{x}^* - \bar{x}^{(0)} \|$. Если взять λ такое, что $0 < \lambda < \dfrac{\xi}{\| \bar{x}^* - \bar{x}^{(0)} \|}$, то точка $\bar{x} = \lambda \bar{x}^* + (1-\lambda)\bar{x}^{(0)}$ лежит в ξ-окрестности точки $\bar{x}^{(0)}$ и $f(\bar{x}) > f(\bar{x}^{(0)})$. Но это не возможно, так как f(x) достигает относительного минимума в точке $\bar{x}^{(0)}$. Таким образом, любой относительный минимум является глобальным минимумом для функции f(x). Если же f(x) строго выпуклая, то абсолютный минимум ее на выпуклом множестве достигается в единственной точке (на середине отрезка, на концах которого f(x) имеет равные значения).





5.Постановка задачи оптимизации.Классы оптимизационных задач:задачи безусловной оптимизации,условной оптимизации,классические на условный экстремум,выпуклые задачи оптимизации,задачи математического программирования.

Задано множество X и функция $f(\bar{x})$,определенная на X.Требуется найти точки max или min функции $f(\bar{x})$ на X.Будем записывать задачу на минимум в виде:

$$f(\bar{x}) \to \min, \ \bar{x} \in X, \text{ где}$$

$f(\bar{x})$ - целевая функция; критерий качества; критериальная функция;
X – допустимое множество; решение; план;
$\forall \bar{x} \in X$ – допустимая точка задачи;
Считается что $X \subset E^n$ т.е. задача является конечномерной.

Точка $\bar{x}^* \in X$ называется точкой глобального минимума функции $f(\bar{x})$ на множестве X, или глобальным решением задачи, если $f(\bar{x}^*) \le f(\bar{x})$ при всех $\bar{x} \in X$.

$\bar{x}^* \in X$ называется точкой локального минимума $f(\bar{x})$ на множестве X, или локальным решением задачи, если существует число $\varepsilon > 0$,такое что:

$f(\bar{x}^*) \le f(\bar{x})$ при всех $\bar{x} \in X \cap U_\varepsilon(\bar{x}^*)$, где

$U_\varepsilon(\bar{x}^*) = \{\bar{x} \in E^n \mid \|\bar{x} - \bar{x}^*\| \le \varepsilon\}$ - шар радиуса $\varepsilon > 0$ с центром в $\bar{x}^*$.

Если указанные ваше неравенства строгие при $\bar{x} \ne \bar{x}^*$,то $\bar{x}^*$ называется точкой строгого минимума, т.е.строгим решением, в глобальном или локальном смысле.

Итак, понятно что $f(\bar{x}^*) = \min f(\bar{x})$, $\bar{x} \in X$ и $\bar{x}^* = \arg\min f(\bar{x})$, $\bar{x} \in X$. Т.е. точка $\bar{x}^*$ реализует величину $f(\bar{x}^*) = f^* = \min f(\bar{x})$, $\bar{x} \in X$. Множество всех точек глобального минимума $f(\bar{x})$ на X обозначается через $\operatorname{Arg}\min f(\bar{x}) = \{\bar{x}^* \in X \mid f(\bar{x}^*) = f^*\}$, $\bar{x} \in X$.

т.е. $\arg\min_{\bar{x} \in X} f(\bar{x}) \in \operatorname{Arg}\min_{\bar{x} \in X} f(\bar{x})$.

Аналогично, для задачи максимизации записываем: $f(\bar{x}) \to \max, \ \bar{x} \in X$. Ясно что такая задача эквивалентна задаче $-f(\bar{x}) \to \min, \ \bar{x} \in X$ в случаях совпадения множеств глобальных и локальных решений этих задач.
Для задач оптимизации вопрос о существовании решения решает Теорема Вейерштрасса:
Пусть X – замкнутое ограниченное множество в E^n.
$f(\bar{x})$ -непрерывная функция на X.Тогда точка глобального минимума функции $f(\bar{x})$ на X существует.
Наиболее важные для теории и приложений объекты оптимизации:

а)Реальные – реальный объект, на котором решается задача оптимизации; необходимо чтобы характеристики объекта изменялись значительно медленней, чем происходит процесс оптимизации, иначе это будет экстремальное регулирование.
б)Модельные – задачу оптимизации решают на физической модели объекта.
в)Математические – математические объекты, для которых критериальная функция и множество X заданы математическими выражениями.

Классы оптимизации задач:
1)Задача безусловной оптимизации

$f(\bar{x}) \to \min, \quad \bar{x} \in E^n$, т.е. здесь $X = E^n$.

2)Задача условной оптимизации

$f(\bar{x}) \to \min, \quad \bar{x} \in X, \quad X \subset E^n$, т.е. X – собственное подмножество пространства E^n.

3)Классическая задача на условный экстремум

$f(\bar{x}) \to \min, \quad \bar{x} \in X$, где

$X = \{\bar{x} \in E^n \mid \tilde{g}_i(\bar{x}) = 0, \ i = 1, 2, \dots, m\}$, или в другой записи

$f(\bar{x}) \to \min, \ \tilde{g}_i(\bar{x}) = 0, \ i = \overline{1, m}$. Здесь явно указывается не само допустимое множество, а система, его определяющая.

4)Выпуклая задача оптимизации

$f(\bar{x}) \to \min, \quad \bar{x} \in X$, где $f(\bar{x})$ - выпуклая функция, заданная на выпуклом множестве X.

5)Задача математического программирования(новейший класс задач на условный экстремум)

$f(\bar{x}) \to \min, \quad \bar{x} \in X$, где

$X = \{\bar{x} \in P \mid \tilde{g}_i(\bar{x}) \le 0, \ i = \overline{1, k}; \ \tilde{g}_i(\bar{x}) = 0, \ i = \overline{k+1, m}\}, \ P \subset E^n$.

Или в такой записи

$f(\bar{x}) \to \min,$

$\begin{cases} \tilde{g}_i(\bar{x}) \le 0, \ i = \overline{1, k}; \\ \tilde{g}_i(\bar{x}) = 0, \ i = \overline{k+1, m}, \end{cases}$ - функциональные ограничения.

$\{ x_j \ge 0, \ j = \overline{1, s}, \ s \le n, \ (\bar{x} = (x_1, x_2, \dots, x_n)^T) \}$ - прямые ограничения. Обычно в качестве P рассматривается множество простой структуры, например координатный параллелепипед $P = \{\bar{x} \in E^n \mid a_j \le x_j \le b_j, \ j = \overline{1, n}\}$.

Если в такой задаче $f(\bar{x}), \ \tilde{g}_i(\bar{x}), \ X$ - выпуклы, то это задача выпуклого программирования.

6)Задачи линейного программирования

$f(\bar{x}) = (\bar{c}, \bar{x}) \to min$, $A\bar{x} = \bar{b}$, $\bar{x} \geq 0$, где

$\bar{c} = \begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{pmatrix}$, $\bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$; $\bar{b} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}$ $A = \begin{pmatrix} a_{11} \dots a_{1n} \\ \dots \\ a_{m1} \dots a_{mn} \end{pmatrix}$

Здесь X – полиэдр, т.е. $X = \{\bar{x} \in P \mid A\bar{x} = \bar{b}\}, P \subset E^n$, , или в развернутом виде

$\sum_{j=1}^{n} c_j x_j \to min$, $\sum_{j=1}^{n} a_{ij} x_j = b_i$, $i = \overline{1, m}$; $x_j \geq 0$, $j = \overline{1, n}$;

существуют и другие формы записи задач линейного программирования.

6.Классы задач оптимизации: линейного программирования с примерами,квадратичного программирования,дискретного программирования,оптимального управления.

1)Задачи квадратичного программирования

$f(\bar{x}) = (\bar{c}, \bar{x}) + (D\bar{x}, \bar{x}) \to min, A\bar{x} = \bar{b}, \bar{x} \geq 0$

Здесь X – полиэдр, т.е. множество, задаваемое линейными условиями. Здесь D – симметрическая неотрицательно-определенная матрица размером n x n, т.е.

$D = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{pmatrix}$, вектор $\bar{c} \in E^n$.

2)Задачи дискретной оптимизации(дискретного программирования)

$f(\bar{x}) \to min, \quad \bar{x} \in X$, где

$X = \{\bar{x} \in D \mid g_i(\bar{x}) \leq 0, i = \overline{1,k}; g_i(\bar{x}) = 0, i = \overline{k+1,m}\}$, где

$D = \overline{D_1 \times D_2 \times \dots \times D_n}$, причем $D_j \subset \{0, \pm 1, \pm 2, \dots\}, j \in J_*^{(1)}$ $D_j \subset R, j \in J$.

Здесь $J_*^{(1)}$ - некоторое подмножество множества $\{1, 2, \dots, n\} = J = J^{(1)} \cup J^{(2)}$.

Если $J^{(1)} = \{1, 2, \dots, n\}$, то имеем задачу целочисленного программирования.

3)Задачи оптимального управления

Постановка этих задач сложнее постановки рассмотренных ранее, поэтому рассмотрим сначала содержательный пример.

Задача о строительстве дороги

Требуется проложить дорогу по неровной местности между двумя пунктами. Затраты на строительство пропорциональны количеству завезенного и вывезенного с трассы грунта. Пусть T – длина дороги, c(t)- известная высота местности в точке на расстоянии $t \in [0,T]$ от начального пункта трассы. Определить функцию $x(t)$, описывающую высоту дороги в каждой её точке $t \in [0,T]$, при которой затраты на её строительство минимальны. При этом наклон(уклон) дороги в любой точке трассы не должен быть больше b_1, т.е. $|\dot{x}(t)| \leq b_1$ для $t \in [0,T]$. Скорость изменения наклона дороги не должна превышать константы b_2, т.е. $|\ddot{x}(t)| \leq b_2$ для $t \in [0,T]$

Уровень дороги в начальном и конечном пунктах определяется выражениями $x(0) = a, \; x(T) = b$.Тогда математическая формулировка задачи такова: минимизировать $\int_0^T |x(t) - c(t)| \, dt$ при условиях

$|\dot{x}(t)| \le b_1, \quad t \in [0,T],$

$|\ddot{x}(t)| \le b_2, \quad t \in [0,T],$

$x(0) = a,$

$x(T) = b.$

Параметром управления здесь является объем грунта, вывезенный или завезенный в точку трассы на расстоянии t от начального пункта, т.е. величина, пропорциональная $|x(t) - c(t)|$.

Перейдем теперь к общей постановке задачи оптимального управления. Движение управляемого объекта описывается системой дифференциальных уравнений

$\dot{\bar{x}} = \bar{f}(\bar{x}, \bar{u}, t)$, $\left(\frac{d\bar{x}}{dt} = f(\bar{x}, \bar{u}, t)\right.$, где $\bar{x} = (x_1, x_2, \dots, x_n)^T$ - где $x_1(t), \dots, x_n(t)$ - n-мерный вектор координат расстояния объекта, или фазовые координаты.

$\bar{u}(t) = (u_1(t), u_2(t), \dots, u_n(t))^T$ - n-мерный вектор управления. t- время.

$\bar{f} = (f_1, f_2, \dots, f_n)$ - n-мерный вектор функций.

Движение объекта подчинено начальным условиям: $\bar{x}(t_0) \in S_0(t_0)$,конечным условиям $\bar{x}(T) \in S(T)$, и ограничениям на фазовые координаты(фазовым ограничениями) $\bar{x}(t) \in X(t), \quad t \in [t_0, T]. \quad (t \in (t_0, T])$.

Кроме того, заданы ограничения на управление: $\bar{u}(t) \in U(t), \ t \in [t_0, T].$ $(t \in (t_0, T])$. Здесь $[t_0, T]$ - отрезок времени, на котором происходит управление системой. $S_0(t_0), S(T), X(t), U(t)$ при каждом t - заданные множества из пространств соответствующих размерностей.

Итак, в этих задачах в качестве элементов, на которых ведется минимизация, выступают функции. Эти задачи относятся к классу задач оптимизации в бесконечномерных функциональных пространствах. Итак, общая модель такова:

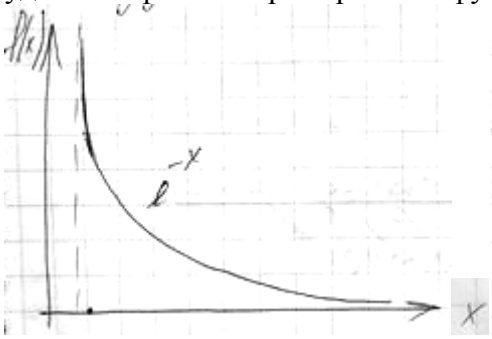
$$I(\bar{u}) = \int_{t_0}^{T} f_0(\bar{x}(t), \bar{u}(t), t)\,dt + \Phi(\bar{x}(T), T) \to \min$$

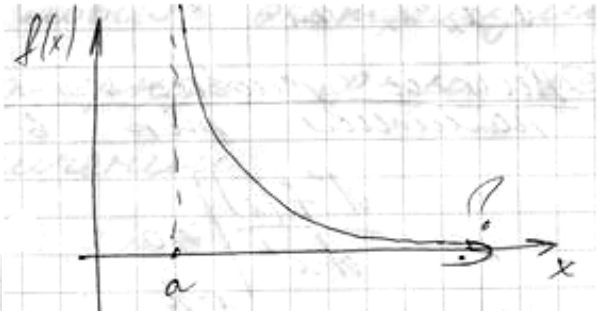
$$\dot{\bar{x}} = \bar{f}(\bar{x}, \bar{u}, t),$$
$$\bar{x}(t_0) \in S_0(t_0),$$
$$\bar{x}(T) \in S(T),$$
$$\bar{x}(t) \in X(t), \quad t \in [t_0, T],$$
$$\bar{u}(t) \in U(t), \quad t \in [t_0, T].$$

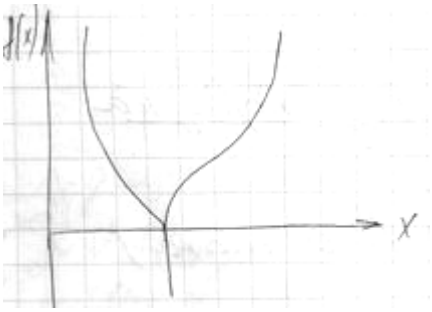
Здесь функционал – это аналог функции цели в конечномерных задачах.

7.Условия экстремума одномерных функций без ограничений.

Стандартные определения локального и глобального решений достаточно нечеткие, и не удовлетворяют например таким функциям: $\min f(x),\ x \in X \subset E$





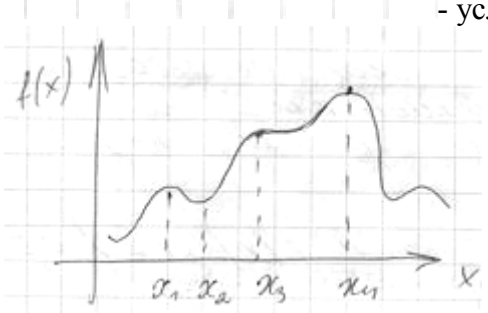


Теорема Вейерштрасса(достаточное)
Любая непрерывная функция, заданная на замкнутом ограниченном множестве достигает своего минимума во внутренней или граничной точке множества.

Необходимое условие I-го порядка(минимум функции в точке $\tilde{x}$)

$$\left.\frac{df(x)}{dx}\right|_{x=\tilde{x}} = 0$$

- условие стационарности.



Необходимое условие II-го порядка(наличие min в точке $\tilde{x}$)

$$\left.\frac{d^2 f(x)}{dx^2}\right|_{x=\tilde{x}} \geq 0$$

Достаточное условие II-го порядка(наличие min в точке $\tilde{x}$)

$$\left.\frac{d^2 f(x)}{dx^2}\right|_{x=\tilde{x}} > 0 \quad (<0 \text{ макс.})$$

Бывает что 2-ая производная = 0, тогда используют аппарат старших производных:

Допустим что в $\tilde{x} \exists$ k производных функции $f(x)$, причем

$f'(\tilde{x}) = f''(\tilde{x}) = \ldots = f^{(k-1)}(\tilde{x}) = 0$

$f^{k}(\tilde{x}) \neq 0$. Если k четное и

$f^{(k)}(\tilde{x}) > 0$,то в точке $\tilde{x}$ находится локальный минимум функции; если k нечетное,то ничего сказать нельзя.

8.Условия экстремума многомерных функций без ограничений. Вид знакоопределенности квадратичной формы.

$f(\bar{x}) \to \min$, $\bar{x} \in X = E^n$ - Необходимое условие

$\left.\frac{\partial f(\bar{x})}{\partial x_j}\right|_{\bar{x}=\tilde{x}} = 0, \; j = \overline{1,n}$

$\operatorname{grad} f(\tilde{x}) = 0;$

$\nabla f(\tilde{x}) = 0$

$f'(\tilde{x}) = 0$

$$H = \begin{pmatrix} \frac{\partial^2 f(\bar{x})}{\partial x_1^2} & \frac{\partial^2 f(\bar{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\bar{x})}{\partial x_1 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f(\bar{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\bar{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\bar{x})}{\partial x_n^2} \end{pmatrix}_{n \times n}$$

$\tilde{x}$ - стационарная точка.

Матрица Гёссе:

Если H неотрицательно определена, т.е. положительно определена или положительно полуопределена, то в исследуемой точке функция является строго выпуклой или просто выпуклой. Если H неположительно определена, т.е. отрицательно определена или отрицательно полуопределена, то в точке функция строго вогнутая или просто вогнутая.

Рассмотрим общий случай: $f(\bar{x}) \to \min$, $\bar{x} \in X = E^n$

$\bar{x}^{(0)} \in X$ и разложение в ряд Тейлора эту функцию в точке $\bar{x}^{(0)}$:

$f(\bar{x}) = f(\bar{x}^{(0)}) + \nabla f(\bar{x}^{(0)})^T \Delta \bar{x} + \frac{1}{2} \Delta x^T \nabla^2 f(\bar{x}^{(0)}) \cdot \Delta \bar{x} + O(\Delta \bar{x})^2$

$\Delta \bar{x} = \bar{x} - \bar{x}^{(0)}$, $O(\Delta \bar{x})^2 \to 0$ при $\Delta \bar{x} \to 0$

Перенесем $f(\bar{x}^{(0)})$ в левую часть:

$\Delta f(\bar{x}) = \nabla f(\bar{x}^{(0)})\Delta\bar{x} + \frac{1}{2}\Delta x^T \nabla^2 f(\bar{x}^{(0)})\Delta\bar{x}$ при $\Delta\bar{x} \to 0$

$\Delta f(\bar{x}) = \frac{1}{2}\Delta\bar{x}^T H \Delta\bar{x}$. Если мы перенесем начало координат в точке $\bar{x}^{(0)}$:

$\Delta f(\bar{x}) = \frac{1}{2}\bar{x}^T H \bar{x}$. $Q(\bar{x}) = \bar{x}^T H \bar{x}$ - квадратичная форма.
Поведение функции полностью определено знакоопределением H.

Для произвольной $A_{m\times n}$:

Угловыми главными минорами $\det|a_{11}|, \det_2\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$ будем обозначать $\Delta_1, \Delta_2, \Delta_3 \dots$.Просто главные миноры D_1, D_2, D_3 (получаются при вычёркивании одноименной строки и столбца)

$D_2: \det\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}, \det\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} \det\begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix}$

Матрица Гёссе положительно определена только если все $\Delta_j > 0, \quad j = \overline{1,n}$
Матрица H положительно полуопределена только если

$D_1 > 0, \quad D_2 > 0, \dots D_r > 0$

где r - ранг H, а $D_{r+1} = \dots = D_n = 0$
Матрица H отрицательно определена только если в ряду

$1, \Delta_1, \Delta_2, \Delta_3, \dots, \Delta_n$ наблюдается строгое чередование знаков.
Матрица H отрицательно полуопределена только если в ряду

$1, D_1, D_2, \dots, D_r$ наблюдается наблюдается строгое чередование знаков,

где r -ранг, а $D_{r+1} = \dots = D_n = 0$

Матрица H является неопределенной если в ряду $1, \Delta_1, \Delta_2, \dots, \Delta_n$ нет строго чередования знаков, но отрицательные элементы присутствуют.

9.Классическая задача условной оптимизации, метод неопределенных множителей Лагранжа(необходимые условия экстремума)

Задача условной оптимизации:

(1) $f(\bar{x}) \to \min$

(2) $\tilde{g}_i(\bar{x}) = 0, \quad i = \overline{1,m}$

$\bar{x} \in X = \{ \bar{x} \mid \tilde{g}_i(\bar{x}) = 0, \ i = \overline{1,m} \}$

$\tilde{g}_i(\bar{x}) = b_i, \quad i = \overline{1,m}$

Метод неопределенных множителей Лагранжа

Необходимо $\mathcal{Y} \neq 0$ (якобиан). Точка $\tilde{\bar{x}}$ - полный дифференциал функции в этой точке = 0.

(3) $df(\tilde{\bar{x}}) = \underbrace{\sum_{j=1}^{m} \frac{\partial f(\tilde{\bar{x}})}{\partial x_j} dx_j}_{\text{ЗП}} + \underbrace{\sum_{j=m+1}^{n} \frac{\partial f(\tilde{\bar{x}})}{\partial x_j}}_{\text{НЗП}} = 0$

Эти дифференциалы связаны следующими зависимостями, которые получаются из (2), если разложить их в ряд Тейлора и ограничиться свободным членом I-го порядка.

(4) $\sum_{j=1}^{m} \frac{\partial g_i(\tilde{\bar{x}})}{\partial x_j} dx_j + \sum_{j=m+1}^{n} \frac{\partial g_i(\tilde{\bar{x}})}{\partial x_j} dx_j = 0, \quad i = \overline{1,m}$

Домножим любое выражение из (4) на неопределенный множитель λ_i и затем (3)+(4).

(5) $\sum_{j=1}^{m} \left(\frac{\partial f(\tilde{\bar{x}})}{\partial x_j} + \lambda_1 \frac{\partial g_1(\tilde{\bar{x}})}{\partial x_j} + \ldots + \lambda_m \frac{\partial g_m(\tilde{\bar{x}})}{\partial x_j} \right) dx_j +$

$+ \sum_{j=m+1}^{n} \left(\frac{\partial f(\tilde{\bar{x}})}{\partial x_j} + \lambda_1 \frac{\partial g_1(\tilde{\bar{x}})}{\partial x_j} + \ldots + \lambda_m \frac{\partial g_m(\tilde{\bar{x}})}{\partial x_j} \right) dx_j = 0, \quad j = \overline{1,m}$

Чтобы оптимальное решение находилось в точке $\tilde{\bar{x}}$ необходимо выполнение условия (5)

(6) $\frac{\partial f(\tilde{\bar{x}})}{\partial x_j} + \lambda_1 \frac{\partial g_1(\tilde{\bar{x}})}{\partial x_j} + \ldots + \lambda_m \frac{\partial g_m(\tilde{\bar{x}})}{\partial x_j} = 0, \quad j = \overline{1,m}$

Система (6) имеет единственное решение в точке $\tilde{\bar{x}}$: $\lambda_1, \lambda_2, \ldots, \lambda_m$.Это следует из того, что якобиан этих функций ограничений $\neq 0$. К системе (6) мы должны добавить необходимое условие

(7) $\frac{\partial f(\tilde{\bar{x}})}{\partial x_j} + \lambda_1 \frac{\partial g_1(\tilde{\bar{x}})}{\partial x_j} + \ldots + \lambda_m \frac{\partial g_m(\tilde{\bar{x}})}{\partial x_j} = 0 \quad j = \overline{m+1,n}$

Для решения задачи (1)-(2) формируется функция Лагранжа:

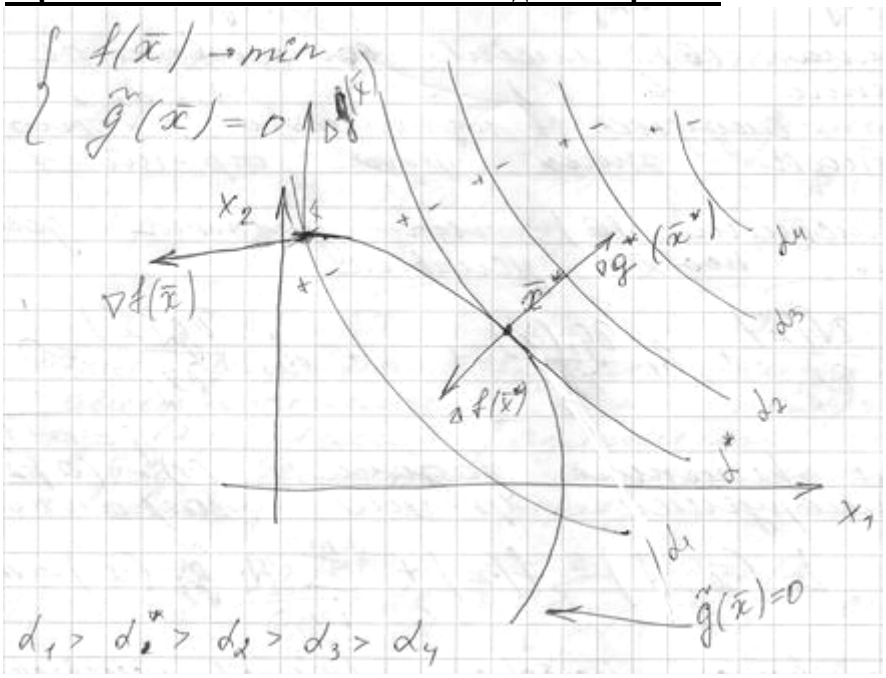
(8) $L(\bar{x}, \bar{\lambda}) = f(\bar{x}) + \sum_{i=1}^{m} \lambda_i \tilde{g}_i(\bar{x}) \to \min$. Для всего набора (n+m) неизвестных выписывается условие стационарности:

(9) $\frac{\partial L}{\partial x_j} = \frac{\partial f(\bar{x})}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i(\bar{x})}{\partial x_j} = 0, \quad j = \overline{1,n}$

(10) $\frac{\partial L}{\partial \lambda_i} = \tilde{g}_i(\bar{x}) = 0, \quad i = \overline{1,m}$

Решая (9)-(10) находим набор стационарных точек для исходных функций.

10.Геометрическая интерпретация множителей и метода Лагранжа, достаточные условия экстремума, седловые точки, решение задач с ограничениями-неравенствами классическим методом Лагранжа



Задача (8)(9)(10) дает возможность найти стационарные точки. Привлекаем аппарат вторых производных для анализа поведения функции в этих стационарных точках.

$$\nabla f(\bar{x}^*) = -\lambda \nabla \tilde{g}(\bar{x}^*);$$

$$\begin{cases} \nabla f(\bar{x}^*) + \lambda \nabla \tilde{g}(\bar{x}^*) = 0 \\ \tilde{g}(\bar{x}^*) = 0 \end{cases}$$

В любых $\bar{\tilde{x}}$ $\frac{\partial L}{\partial x_j} = 0$, $\frac{\partial L}{\partial \lambda_i} = 0$, численно $L(\bar{x}, \bar{\lambda}) = f(\bar{x})$ в каждой такой точке. Нас интересует изменение значения функции Лагранжа при изменении $\bar{x}_\delta$:

$$\Delta f(\bar{x}) = \Delta L(\Delta x_j, \bar{\lambda}) = \sum_{j=\overline{1,n}} \frac{\partial L(\bar{x},\bar{\lambda})}{\partial x_j} \Delta x_j + \frac{1}{2} \Delta \bar{x}^T H(L) \Delta \bar{x} + o(\Delta \bar{x})^2 \quad ; \Rightarrow$$

$\frac{\partial L(\bar{x},\bar{\lambda})}{\partial x_j} = 0$

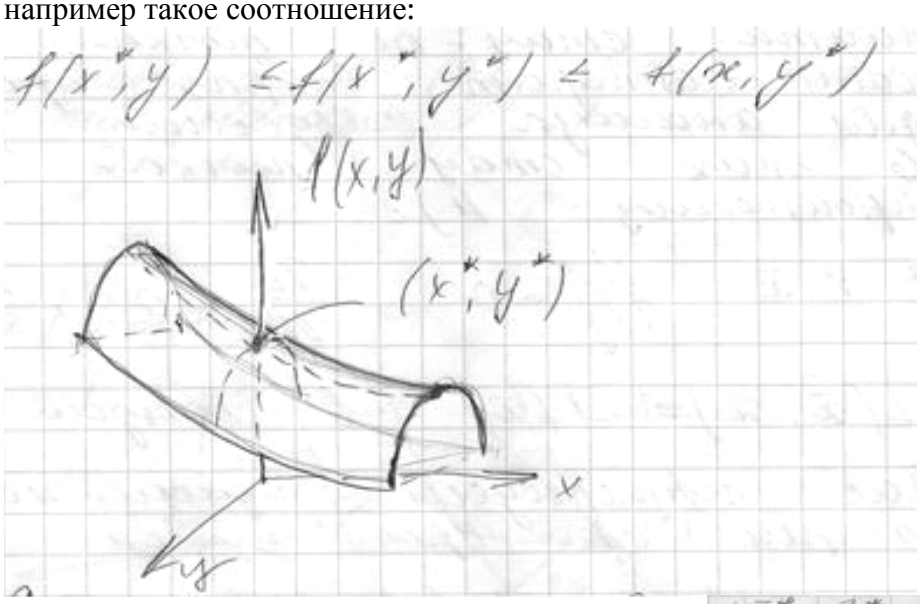
$\to 0, \Delta \bar{x} \to 0$

$$\Delta L(\Delta \bar{x}, \bar{\lambda}) = \frac{1}{2} \Delta \bar{x}^T H(L) \Delta \bar{x};$$

$$Q(\bar{x}) = \bar{x}^T H(L) \bar{x}$$

Точка (x^*, y^*) называется <u>седловой точкой</u> функции если для неё выполняется например такое соотношение:

$$f(x^*, y) \le f(x^*, y^*) \le f(x, y^*)$$



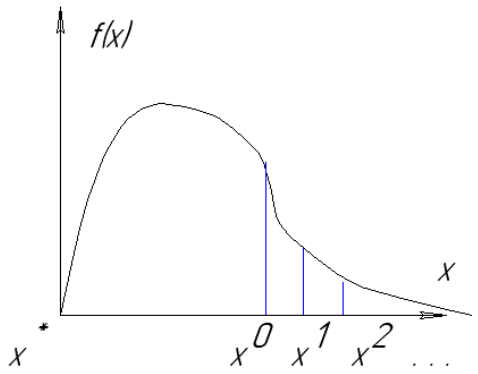
<u>Стратегическая седловая точка функции Лагранжа</u> $(\bar{x}^*, \bar{\lambda}^*)$, если выполняется $L(\bar{x}^*, \bar{\lambda}) \le L(\bar{x}^*, \bar{\lambda}^*) \le L(\bar{x}, \bar{\lambda}^*)$.

11. понятие о численных методах оптимизации

В любом численно методе так или иначе выбираются точки в пространстве которые являются последовательными приближениями к x* (решению задачи) В каждой точке выполняются определенные вычисления и получаются результаты y.
Алгоритм считается заданным если

1) задана последовательность точек вычисления
2) определены какие вычисления выполняются в каждой точке
3) задано условие остановки алгоритма

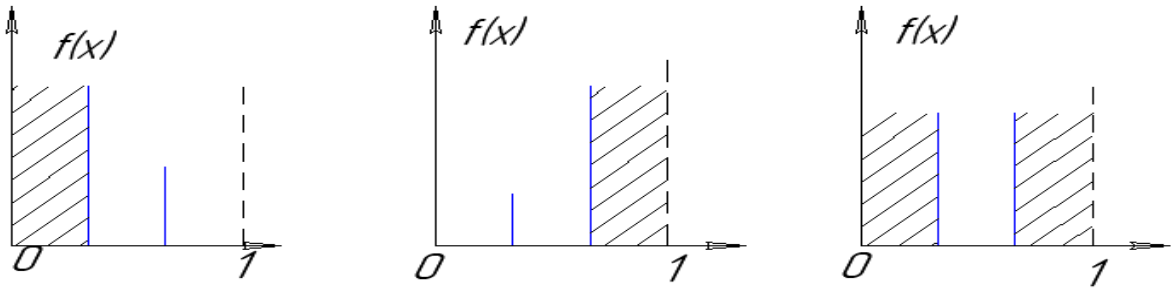
если метод при своей работе использует только значения функции цели и функций ограничений, то такой метод называют методом первого порядка. Если дополнительно используются значения n-ых производный то это метод n-го порядка.
Методы подразделяются на пассивный и последовательный поиск.
В методе пассивного поиска все точки задаются заранее до начала работы алгоритма. При последовательном поиске очередная точка определяется после того, как была определена предидущая.
Любой численный алгоритм задается простой формулой:
$x^{(k+1)} = x^{(k)} + a_k h^{(k)},\ k = 0,1,2..\ a_k = const\quad h^{(k)}$- определяет направление шага, константак a_k определяет длину шага.
Численные методы подразделяются на конечношаговые и безконечношаговые. Конечношаговые – это методы линейного и квадратичного программирования, остальные безконеяношаговые. Важной характеристикой для безконечношаговых методов ябляется сходимость. Говорят что метод сходится если $x^{(k)} -> x^{(*)}\ pri\ k -> \infty$, часто говорят что метод сходится по функции если $f\left(x^{(k)}\right) \to f(x*),\ k \to \infty$. Последовательность точек $x^{(0)}, x^{(1)}, ..., x^{(k)}$ для которых $f\left(x^{(0)}\right) > f\left(x^{(1)}\right) > ... >$ называется минимизирующей (релаксирующей) последовательностью. Сходимость алгоритма часто оценивается с помощью скорости сходимости. Метод сходится линейно (слинейной скоростью) если для него можно подобрать q от (0,1) и некоторое число ko – номер операции при котором выполняется:
$\| x^{(k+1)} - x* \| \le q \| x^{(k)} - x* \|\ pri\ k \ge k_0$



2) говорят что метод сходиься со сверхлинейной скоростью если для него выполняется соотношение: $\| x^{(k+1)} - x* \| \le q_{k+1} \| x^{(k)} - x*\ pri\ q_{k+1} \to 0 +\ \ pri\ k \to \infty$.
3) с квадратичной скоростью если можно подобрать некоторое константу c>0 при которой выполняется соотношение$\| x^{(k+1)} - x* \| \le c \| x^{(k)} - x* \|^2\ \ pri\ k \ge k_0$.
Используются следующие условия остановки работы алгоритма:
1) $\| x^{(k+1)} - x^{(k)} \| \le \varepsilon_1$
2) $\| f\left(x^{(k+1)}\right) - f\left(x^{(k)}\right) \| \le \varepsilon_2$
3) $\| f'\left(x^{(k+1)}\right) \| \le \varepsilon_3$
Эти же условия остановки обычно записываются так:
1) $\| x^{(k+1)} - x^{(k)} \| \le \partial_1 \left(1 - \| x^{(k+1)} \|\right)$
2) $\| f\left(x^{(k+1)}\right) - f\left(x^{(k)}\right) \| \le \delta_2 \left(1 - \| f\left(x^{(k+1)}\right) \|\right)$
3) $\| f'\left(x^{(k+1)}\right) \| \le \delta_3 \left(1 + \| f'\left(x^{(k+1)}\right)\right)$
Точка x* является локально устойчивой если к ней сходится любая минимизирующая последовательность.

Количественной характеристикой качества решаемой задачи является число обусловленности: $\mu = \lim_{\delta \to 0} \left(\sup_{x \in L\delta} \| x - x^* \|^2 \Big/ \inf_{x \in L\delta} \| x - x^* \|^2 \right)$ где $L\delta = \{x \mid f(x) - f(x^*) = \delta$, понятно что μ >=1 если μ примерно равно 1 то говорят что задача хорошо обусловлена, т.е ее линии уровня представляют собой концентрические окружности. Если μ >>1 то линии уровня сильно вытянулы в одних направлениях и слабо в других.

12. одномерный пассивный поиск.
Унимодальные функции(имеющие единственный минимум на заданном интервале). Вычисление значения функции в точке называется эксперимент. По результатам любой пары экспериментов на исходном интервале можно сузить исходный интервал и возможны 3 исхода:



Эффективность поиска определяется стратегией расположения экспериментов.
Интервалом неопределенности называетмя интервал возможных значений x в которых с гарантией находится точка минимума функции.
Пусть для f(x) выполнено N экспериментов x1,x2,…,xn олгда длинна интервала неопределнности рассчитывается: 1) $x_k : f(x_k) = \min_{1 \le i \le n} f(x_i)$ 2) $l(x1, x2, ..., xn, k) = x_{k+1} - x_{k-1}$

Пессимистическая оценка длинны интервала неопределнности:
$L_n = L(x1, x2, ..., xn) = \max_{1 \le k \le n} l(x1, x2 .... , xn, k)$

Принцип выбора оптимальной стратегии (принцип минимакса) в качестве оптимальной надо выбирать стратегию, приводит к минимуму значению максимальной длинны интервала неопределнности т.е к минимуму значения пессимистической оценки длинны интрвала неопределенности. $\hat{L}_n$-оптимальная стратегия

поиска. $\hat{L}_n = \hat{L}(\hat{x}1, ..., \hat{x}n) = \inf_{0 \le x1 ... \le xn \le 1} L(x1, ..., xn) = \inf_{0 \le x1 ... \le xn \le 1} \max_{1 \le k \le n} l(x1, ..., xn, k)$ эта стратегия не может дать результата хуже чем $\hat{L}_n$, а скорее всего даст лучше

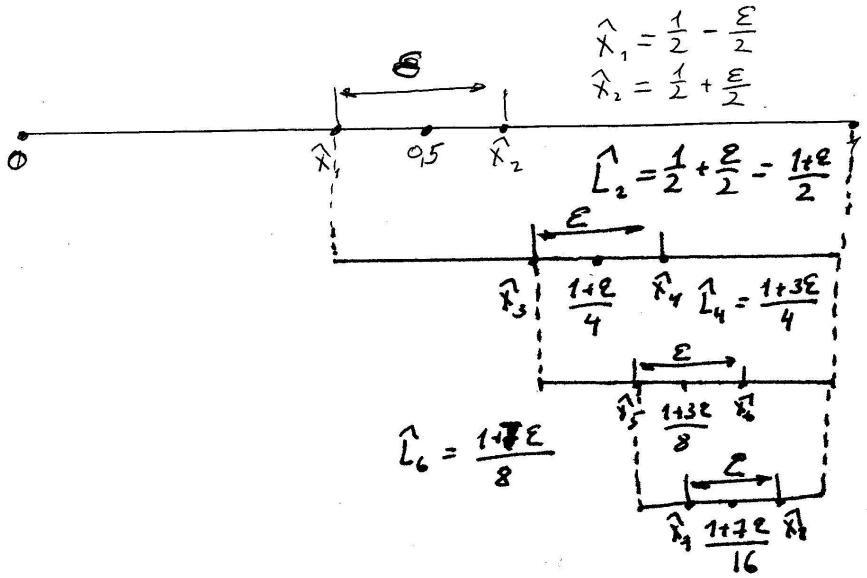
1) n-четное число экспериментов тогда

$\hat{L}_n = \frac{1+\varepsilon}{\frac{n}{2}+1}$; __ $\hat{x}_k = \frac{(1+\varepsilon)\left[\frac{k+1}{2}\right]}{\frac{n}{2}+1} - \varepsilon\left\{\left[\frac{k+1}{2}\right] - \left[\frac{k}{2}\right]\right\}$ где […] – выделение целой части а ε - минимально возможно расстояние на котором еще различимы f(x1) И f(x2)

2) (n+1) – нечетное число экспериментов

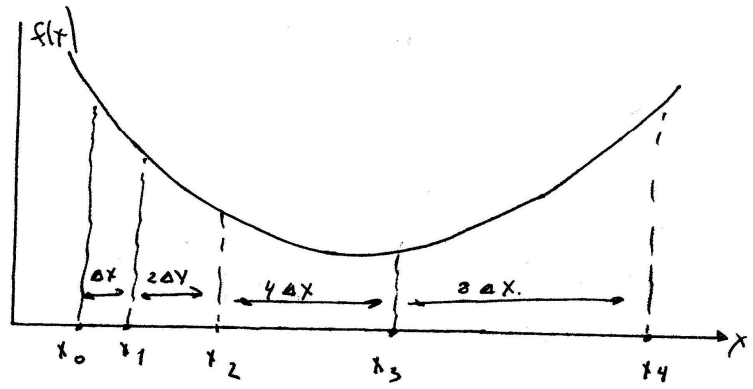
$\hat{L}_{n+1} = \frac{1}{\frac{n}{2}+1}$; $\hat{L}_n - \hat{L}_{n+1} = \frac{1+\varepsilon}{\frac{n}{2}+1} - \frac{1}{\frac{n}{2}+1} = \frac{\varepsilon}{\frac{n}{2}+1}$

13. метод дихотомии метод свена



Эксперименты проводим парами, $\widehat{L}_n = 2^{-\frac{n}{2}} + \varepsilon\left(1 - 2^{-\frac{n}{2}}\right)$

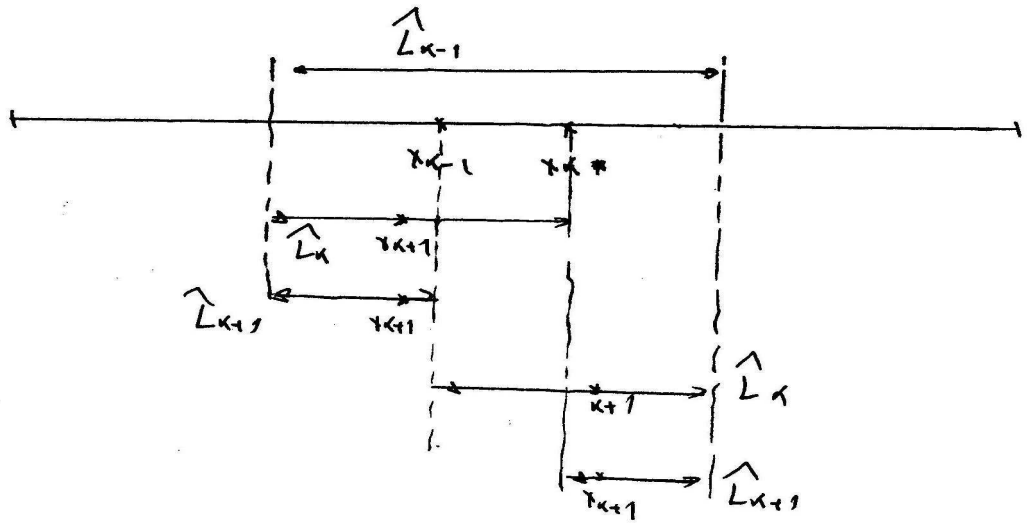
2) метод свена (медод грубого поиска или метод удвоения интервалов) предназначен для первоначального сужения исходного интервала перед применением более сложных методов поиска.



$\widehat{x}_{k+1} = \widehat{x}_k + \Delta x * 2^k;\ k = 0,1,2$

14. метод Фибоначчи метод золотого сечения.

1)метод Фибоначчи: в этом методе на первой итерации проводится 2 эксперимента а на каждой последующей итерации добавляется 1 эксперимент, причем эксперименты располагаются симметрично на расстоянии <0.5 $\widehat{L}_{tek}$ на последней итерации два последних эксперимента располагаются на расстоянии ε. Заранее рассчитывается n исходя из точности ε



$L_{k-1} = L_k + L_{k+1}$ *или* $L_{j-1} = L_j + L_{j+1};\ j = \overline{2, n-1}$

Расчитаем длины интервалов от конца:

$k = 1,\ldots, n-1$

F *– числа фиббоначи*

$F_0 = F_1 = 1$

$F_k = F_{k-2} + F_{k-1}$

$L_n = L_n$

$L_{n-1} = 2L_n - \varepsilon$

$L_{n-2} = L_{n-1} + L_n = 3L_n - \varepsilon$

$L_{n-3} = L_{n-2} + L_{n-1} = 5L_n - 2\varepsilon$

$L_{n-4} = L_{n-3} + L_{n-2} = 8L_n - 3\varepsilon$

$L_{n-k} = F_{k+1}L_n - F_{k-1}\varepsilon$

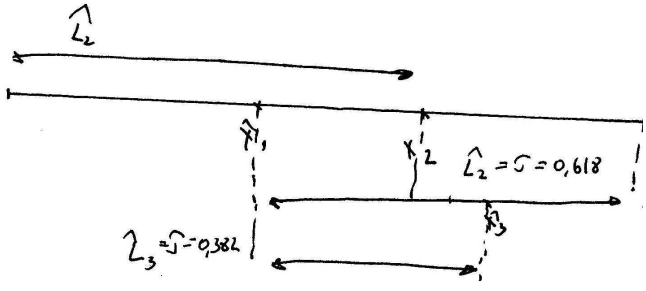
$L_n = \dfrac{1}{F_n} + \dfrac{F_{n-2}}{F_n}\varepsilon$ n находим из $F_{n+1} < \dfrac{b_0 - a_0}{\varepsilon} \le F_{n+2}$

2) метод золотого сечения.

n-заранее не определяется $L_{j-1} = L_j + L_{j+1}$ - единственное начальное условие

$$\frac{L_{j-1}}{L_{j-1}} = \frac{L_j}{L_{j-1}} + \frac{L_{j+1}}{L_{j-1}} <=> 1 = \tau + \tau^2 => \tau = 0.618 \quad \tau^2 = 1 - \tau = 0.382$$

$L_n = \tau^{n-1}$



Метод Фибоначчи и метод золотого сечения обладают линейной скоростью сходимости.

15. методы оценивания с помощью квадратичной аппроксимации.
1)метод однократного оценивания: f(x)- унимодальная непрерывная достаточно гладкая, тогда ее в ограниченном интервале можно аппроксимировать некоторой квадратичной функций а затем искать минимум этой квадратичной функции.
Есть 3 точки: x1, x2, x3 соответственно f1, f2, f3 потребуем чтобы в этих точках значение $\varphi(x)$ совпадало с f(x)

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

$$f(x_1) = \varphi(x_1) = f1 = a_0; \quad a_0 = f_1$$

$$f(x_2) = \varphi(x_2) = f2 = f1 + a_1(x - x_1); \; a1 = \frac{f_2 - f_1}{x_2 - x_1}$$

$$f(x_3) = \varphi(x_3) = f_3 = f1 + \frac{f_2 - f_1}{x_2 - x_1}(x_3 - x_1) + a2 * (x_3 - x_1)(x_3 - x_2) =>$$

$$=> a_2 = \frac{f_3 - f_1}{(x_3 - x_1)(x_3 - x_2)} - \frac{f_2 - f_1}{(x_2 - x_1)(x_3 - x_2)}$$

$$\varphi'(x) = \frac{d\varphi(x)}{dx} = a_1 + a_2(x - x_2) + a_2(x - x_1) = 0 =>$$

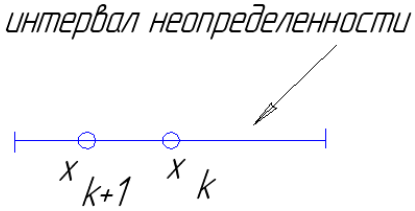
$$\tilde{x} = -\frac{a_1}{2a_2} + \frac{(x_2 + x_1)}{2} \approx x^*$$

2)метод последовательного оценивания (метод пауэла):
а) Предварительный.
Выбираем начальную точку x1 и шаг поиска Δx , рассчитываем f(x1) если
$f(x_2 = x_1 + \Delta x) < f(x_1)$, *То* $x_3 = x_2 + \Delta x$ *иначе* $x_3 = x_1 - \Delta x$
б) переназначаем x1 x2 x3 в естественном порядке и (если надо заного вычисляем f1 f2 f3). Далее выбираем $\min f = \min\{f_1, f_2, f_3\}$ и соответствующий $x_{\min} = \arg(\min f)$ затем по методу однократного оценивания проводим квадратичную аппроксимацию x1 x2 x3 и находим очередную точку приближения $\tilde{x}$
проверяем если $|\min f - f(\tilde{x}) \le \varepsilon_1$ *И* $|x_{\min} - \tilde{x}| \le \varepsilon_2$ то заканчиваем работы и $x_{\min}$ *или* $\tilde{x}$ будут решением, иначе переходим в шаг в)
в) выбираем наилучшую из $x_{\min}$ *или* $\tilde{x}$ и 2 точки по обе стороны от нее(из числа имеющихся) и уходим на шаг б).
<u>замечание:</u> на первой итерации алгоритма точка $\tilde{x}$ может оказаться далеко за x3 или далеко перд x1 и тогда ее невозможно окружить 2-мя точками, тогда надо применить метод свена и используя Δx провести несколько шагов из $\tilde{x}$, затем снова на шаг б)

методы квадратичной аппроксимации обладают суперлинейной скоростью сходимости.

16) метод средней точки метод касательных метод секущих

1) метод деления интервала пополам (метод средней точки):



по знаку $f'(x_k)$ определяем неперспективную половину, в оставшейся находим середину и вычисляем $f'(x_{k+1})$.

Условие остановки: $|f'(x_k)| \le \varepsilon$

2) метод касательных (метод Ньютона): Решение гарантировано если f(x) строго выпукла ($f''(x) > 0$); $f'(x) = 0 \Rightarrow$ *находим* $\tilde{x}$, работа происходит в координатной системе $f'(x), x$.

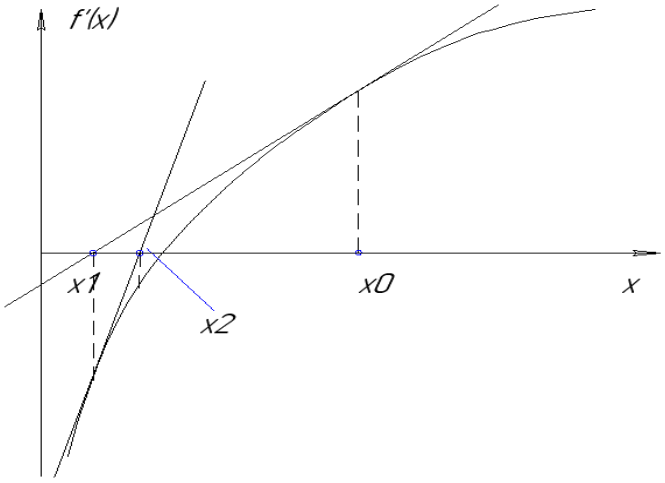
Выберем x0 и выполним начальную аппроксимацию т.е. разложим f '(x) в ряд Тейлора:

$$0 = f'(x) = f'(x_0) + f''(x_0)(x - x_0)$$

$$\tilde{x}_1 = x_0 - \frac{f'(x_0)}{f''(x_0)}; \tilde{x}_2 = x_1 - \frac{f'(x_1)}{f''(x_1)}; \tilde{x}_k = x_{1-1} - \frac{f'(x_{k-1})}{f''(x_{k-1})}$$

Условие остановки: $|f'(\tilde{x}_k)| \le \varepsilon$

Геометрическая интерпретация:



Если аналитического выражения для производной не существует тогда пользуемся разностными выражениями:

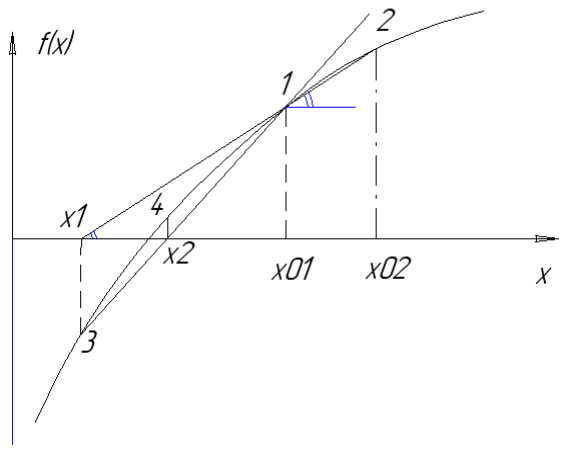
$$f'(x) \cong \hat{f}'(x) = \frac{1}{2g}(f(x+g) - f(x-g)),$$

где g- даза оценки

$$f''(x) \cong \hat{f}''(x) = \frac{1}{4g^2}\left[\begin{matrix} f(x+2g) - 2f(x) + \\ + f(x-2g) \end{matrix}\right]$$

3) метод секущих: более универсальны (не надо искать 2 производную) работает в том числе и на невыпуклых функциях, но для гарантии решения требует чтобы для невыпуклых функций знаки производных были различны в очередных 2-х точках

$f'(x_1) \cdot f'(x_2) < 0$



Запишем уравнение для секущей:

$$0 = f'(x) = \frac{(x - x_0)(f'(x_{02}) - f'(x_{01}))}{(x_{02} - x_{01})} + f'(x_{01}),$$

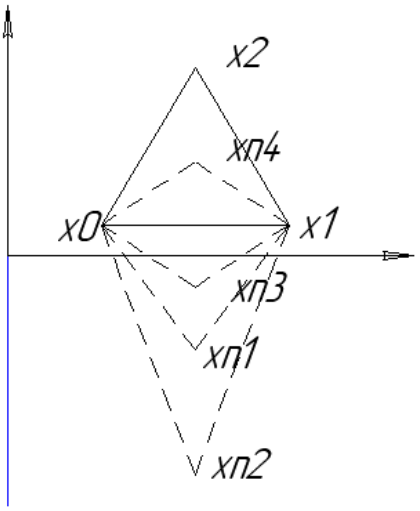
находим $x = x_{01} - \frac{f'(x_{01})(x_{02} - x_{01})}{f'(x_{02}) - f'(x_{01})}$ проводим очередную секущую через точки 3 и 1 если $f'(x_2) > 0$ то очередную хорду проводим через 3 и 4.

условие остановки: $|f'(x_k)| < \varepsilon$

17) метод поиска по симплексу(S^2 *метод*) 1962 год , Нелдер и Мид 1965.

Симплекс – это многогранник в n-мерном пространстве, образованный n+1 вершинами, стоящих на равном расстоянии друг от друга.

Процедура отражения любой вершины симплекса, т.е перенос некоторой вершины симплекса по прямой, проходящей через эту точку и через центр тяжести остальных вершин на некоторое расстояние(т.е используя свойство симплекса что на любой его гране можно построить новый симплекс.



xn1-нормальное отражение, xn2-растяжение симплекса

Применительно к k-ой итерации

1) если надо, переобозначаем вершины $x^{(0)}(k), x^{(1)}(k),..., x^{(n)}(k)$ так чтобы оно удовлетворяло $f\left(x^{(0)}(k)\right) \le f\left(x^{(1)}(k)\right) \le ... \le f\left(x^{(n)}(k)\right)$, при этом n-ая точка является худшей точкой и отражению подвергается всегда худшая точка.
2) Найдем центр тяжести остальных вершин $x^{(c)}(k) = \frac{1}{n}\sum_{i=0}^{n-1} x^{(i)}(k)$

3) Вычислим пробную точку для нормального отображения $\bar{u}(k) = \bar{x}^{(n)}(k) + (1+Q)\left(\bar{x}^{(c)}(k) - \bar{x}^{(n)}(k)\right), Q = \alpha = 1$

Случай α: $f\left(\bar{x}^{(0)}(k)\right) \le f(\bar{u}(k)) \le f\left(\bar{x}^{(n-1)}(k)\right)$ => именно $\bar{u}(k)$ вершиной заменяем $\bar{x}^{(n)}(k)$ и уходим на (k+1) итерацию

Случай β: если $f(\bar{u}(k)) < f\left(x^{(0)}(k)\right)$ то направление перемещения в $\bar{u}(k)$ перспективно и тогда следует проверить случай растяжения симплекса в этом направлении $\bar{v}(k) = \bar{x}^{(n)}(k) + (1+Q)\left(\bar{x}^{(c)}(k) - \bar{x}^{(n)}(k)\right), Q = \beta = 2$ если $f(\bar{v}(k)) < f(\bar{u}(k))$ то $\bar{u}(k)$ удаляем и заменяем ее на $\bar{v}(k)$ иначе заменяем $\bar{x}^{(n)}(k)$ *на* $\bar{u}(k)$ и уходим на начало (к+1) итерации.

Случай γ :если $f(\bar{u}(k)) > f\left(\bar{x}^{(n-1)}(k)\right)$ вычисляем пробную величину $\bar{w}(k) = \begin{cases} \text{если } f(\bar{u}(k)) \le f\left(\bar{x}^{(n)}(k)\right) \text{то } \bar{x}^{(n)}(k) + (1+\gamma)\left(\bar{x}^{(c)}(k) - \bar{x}^{(n)}(k)\right) \\ \text{иначе } \bar{x}^{(n)}(k) + (1-\gamma)\left(\bar{x}^{(c)}(k) - \bar{x}^{(n)}(k)\right) \end{cases}$ $\gamma = 0.5$

Если $f(\bar{w}(k)) < \min\{f(\bar{u}(k)), f\left(\bar{x}^{(n)}(k)\right)\}$ то поиск удачен и вершиной $\bar{w}(k)$ заменяем вершину $\bar{x}^{(n)}(k)$ и уходим на начало (к+1) итерации, иначе равномерно сжимаем симплекс к вершине $\bar{x}^{(0)}(k)$ по формуле $\bar{x}^{(i)}(k) = \bar{x}^{(i)}(k) + 0.5\left(\bar{x}^{(0)}(k) - \bar{x}^{(i)}(k)\right)$ $i = \overline{1,n}$ и уходим в начало (к+1) итерации.

Условие остановки: $\left\{\frac{1}{n}\sum_{i=1}^{n}\left[f\left(\bar{x}^{(0)}(k+1)\right) - f\left(\bar{x}^{(i)}(k+1)\right)\right]^2\right\}^{\frac{1}{2}} \le \varepsilon$

Процедура построения начального симплекса:

$\bar{x}^{(0)}(0) = (x_1, x_2, .., x_n)^T$, *тогда* $\bar{x}^{(1)}(0) = (x_1 + r, x_2 + s, ..., x_n + s); \bar{x}^{(2)}(0) = (x_1 + s, x_2 + r, ..., x_n + s);$

$\bar{x}^{(n)}(0) = (x_1 + s, x_2 + s, ..., x_n + r);$

где $r = \mu\left(\frac{\sqrt{n+1}-1}{n\sqrt{2}}\right)$; $s = \mu\left(\frac{\sqrt{n+1}+n-1}{n\sqrt{2}}\right)$

Если $\mu = 1$ то ребра начального симплекса имеют примерно

единичную длину.

Если некоторая вершина не исключается к М-ой итерации то текущий симплекс заменяется новым с вершиной $\bar{x}^{(0)}$, соответсвующей наилучшей в текущем симплексе.

$M = 1.65n + 0.05n^2$

18) метод поиска Ху-ка Дживса.

Если из $\bar{x}^{(k)}$ выходить и двигаться в направлении $\bar{d} = \left(\bar{x}^{(k)} - \bar{x}^{(k-1)}\right)$, то процесс обычно сходиться быстрее к точке $\bar{x}^*$.

Первая процедура: исследующий поиск. $\bar{x}^{(t)}$ - текущая точка (базовая) выбираются $\Delta x_j;\ j = 1, n$ пробные шаги затем из $\bar{x}^{(t)}$ делаются 2 шага $\pm \Delta x_1$ и фиксируются и фиксируются значения остальных координат, если один из них привел к успеху т.е уменьшению функции, то фиксируемся в этой точке и уже из нее делаем шаги вдоль следующей координаты $\pm \Delta x_2$ в результате находиться точка $x^{(t+1)}$. Эта процедура обеспечивает локальный поиск и движению к дну оырагов.

Вторая процедура: поизк по образцу $\bar{x}^{(k+1)} = x^{(k)} + \left(\bar{x}^{(k)} - \bar{x}^{(k-1)}\right)$, $x^{(k)}$ *и* $x^{(k-1)}$ - базовые точки, производиться движение вдоль оврагов.

Алгоритм:

Подготовительный этап: 1. $\bar{x}^{(0)}, \Delta x_j \quad j = 1, n, \alpha > 1\ \varepsilon > 0$

Шаг 2: в последней из найденных базовой точке $\bar{x}^{(k-1)}$ производим исследующий поиск (в начале это точка $\bar{x}^{(0)}$), если он приводит к успеху то последнюю из успешно найденных точек обозначаем $\bar{x}^{(k)}$ и переходим к 4.

Шаг 3: если $\| \Delta x \| \le \varepsilon$ то остановка последняя из базовых точек является окончательным приближением, иначе уменьшаем шаг $\Delta x_j = \dfrac{\Delta x_j}{\alpha};\ j = 1, n$

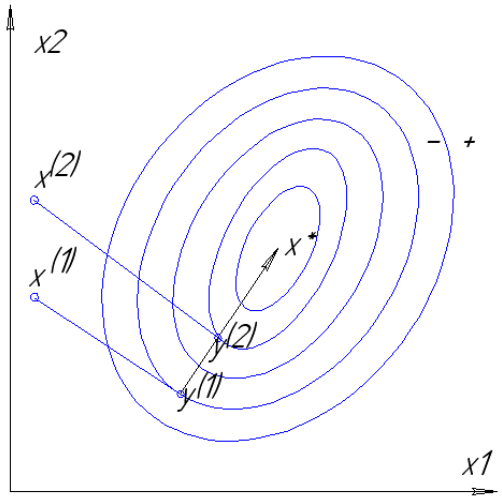
Шаг 4: выполняем поиск по образцу $\bar{x}^{(k+1)}_{врем1} = \bar{x}^{(k)} + \left(\bar{x}^{(k)} - \bar{x}^{(k-1)}\right)$ затем проводим в найденной точке исследующий поиск и получаем $x^{(k+1)}_{врем2}$ и выбираем наилучшую $x^{(k+1)} = \arg\min\left[f\left(x^{(k+1)}_{врем1}\right), f\left(x^{(k+1)}_{врем2}\right)\right]$, если $f\left(x^{(k+1)}\right) < f(x)^{(k)}$, то шаг 4 считаем удачным и уходим на его начало, при этом проводим переобозначение точек $x^{(k+1)} -> x^{(k)}; x^{(k)} -> x^{(k-1)}$, иначе остаемся в последней базовой точке $x^{(k)}$ н обозначаем ее как $x^{(k-1)}$ и уходим на шаг 3.

19. метод сопряженных направлений Пауэла.

Предназначен для минимизации функций: $f(x) = a + b\bar{x} + 0.5\bar{x}^T C\bar{x} - > \min$ не боее чем за n итераций (работает в рациональных дробях)

На первых (n-1) итерациях находится совокупность направлений в пространстве $d^{(1)}, d^{(2)}, ..., d^{(n-1)}$ и на n-ой итерации выполняется последующая минимизация в этих направлениях. Фактически переходим в новую систему координат, оси которой совпадают с осями квадратичной функции $C_{n\times n}$ является диагональной,

$d^{(i)}$ и $d^{(j)}$ называются сопряженными или сопряженными по отношению к матрице С, или С-ортогональными, если выполняется: $d^{(i)\,T} \bullet C \bullet d^{(j)} = 0 \; i, j = 1, n \; i \neq j$



Работает для любых направлений. Докажем что $(\bar{y}^{(2)} - y^{(1)})$ является сопряженным с d

Это свойство для квадратичной функции называется построением параллельног оподпространства.

$\bar{x}^{(1)}, d, f(\bar{x}^{(1)} + \lambda d), y^{(1)}$

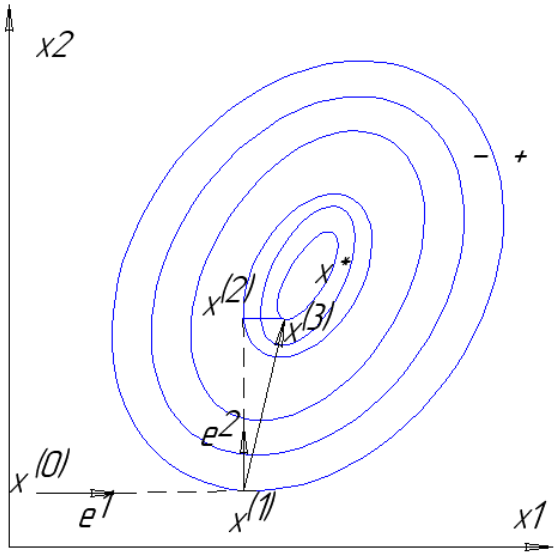
$\bar{x}^{(21)}, d, f(\bar{x}^{(2)} + \lambda d), y^{(2)}$

$$\frac{df}{d\bar{x}}\frac{dx}{d\lambda} = (\bar{b}^T + \bar{x}^T C)\bar{d} = 0$$

$$-(b^T + y^{(2)T} C)\bar{d} = 0$$

$$(b^T + y^{(1)T} C)\bar{d} = 0$$

$$= (y^{(2)} - y^{(1)})C\bar{d} = 0$$



Обычно выходят из одной начальной точки

Алгоритм:

1. задаем начальную точку $x^{(0)}$ и набор линейно независимых ортогональных направлений $S^{(1)}, S^{(2)}, ..., S^{(n)}$
2. строим текущюю линейку направлений поиска(в начале $S^{(1)}, S^{(2)}, ..., S^{(n)}$), затем из последней точки эксперимента делаем (n+1) последовательных шагов минимизации функции в направлениях $S^{(n)}, S^{(1)}, S^{(2)}, ..., S^{(n)}$ и обозначаем точки минимизации $x^{(1)}, x^{(2)}, .., x^{(n+1)}$
3. определяем новое сопряженное направление $d^{(1)} = x^{(n+1)} - x^{(1)}$
4. формируем очередную линейку направления поиска $S^{(1)}, S^{(2)}, ..., S^{(n)}$ и в конец дописываем очередное сопряженное направление и выбрасываем первое $S^{(2)}, ..., S^{(n)}, d^{(1)}$ и уходим на шаг 2

со 2-ой итерации выполняем (n+1) шагов последовательного поиска в направлениях $d^{(1)}, S^{(2)}, ..., S^{(n)}, d^{(1)}$ в результате находим $d^{(2)}$,сопряженное с $d^{(1)}$, после (n-1) итерации алгоритма получаем $S^{(n)}, d^{(1)}, d^{(2)}, ..., d^{(n-1)}$, и теперь рассмотрим n задач последовательной минимизации, в результате получим решение задачи.

Сходимость – суперлинейная.

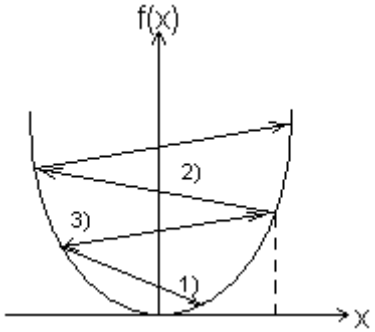
Вопрос 20.
Градиентные методы: с постоянным шагом, с дроблением шага.

Градиентный метод с постоянным шагом.

$$\vec{o}^{(k+1)} = \vec{o}^{(k)} - \alpha f'(\vec{o}^{(k)}), k = 0,1,2...$$

$$x_j^{(k+1)} = x_j^{(k)} - \alpha \frac{\partial f(\frac{1}{x^{(k)}})}{\partial x_j}, \; j = \overline{1,n}$$

$$f(x) = ax^2$$



$\bar{x}^*$ x_0

$$x^{(k+1)} = x^{(k)} - 2\alpha a x^{(k)} = x^{(k)}(1 - 2\alpha a)$$

$$1) 0 < \alpha < \frac{1}{a} - \text{ìåòîä} \quad \text{_ñõîäèòñÿ}$$

$$2) \alpha > \frac{1}{a} - \text{ìåòîä} \quad \text{_ ðàñõîäèòñÿ}$$

$$3) \alpha = \frac{1}{\text{à}} - \text{çàöèêëèâàí èå}$$

Если выбирать малые α, метод надёжно приводит в точку $\bar{x}^*$. Иначе метод может заканчиваться неудачно.

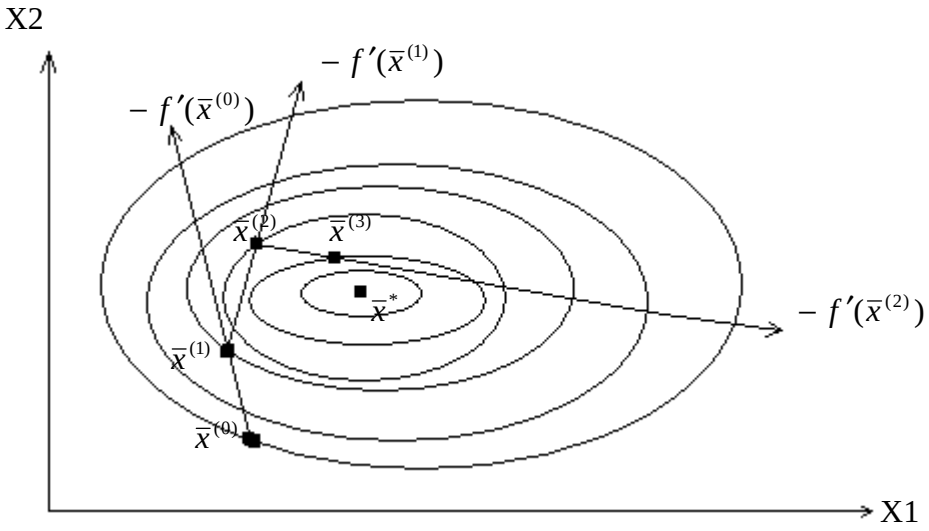
Градиентный метод с дроблением шага.

В методе длина шага постепенно уменьшается по мере приближения к точке $\bar{x}^*$. На каждой итерации шаг выбирается таким образом, чтобы для него выполнялось соотношение:

$$f(\bar{x}^{(k)} - \underbrace{\alpha_k f'(\bar{x}^{(k)})}_{\Delta\bar{x}}) - f(\bar{x}^{(k)}) \le -\varepsilon\alpha_k \left\| f(\bar{x}^{(k)}) \right\|^2$$

$$f(\bar{x}) = f(\bar{x}^{(k)}) + f'(\bar{x}^{(k)})\Delta x$$

Выбирается шаг достаточной величины и проверяется выполнение соотношения. Если выполняется – делаем уменьшение. Если нет – дробим шаг, пока соотношение не начинает выполняться.



$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \alpha_k f'(\bar{x}^{(k)}), k = 0,1,2...$

Если $f(\bar{x})$ такая, что для нее существует постоянная Липшица (для которой: $\|f(\bar{x}) - f(\bar{y})\| \le R\|\bar{x} - \bar{y}\|, \forall x, y \in \mathrm{E}^n$), то $\alpha_k = \alpha = const = \frac{1-\varepsilon}{R}$

Если для матрицы $f''(\bar{x})$ вычислять оценки сверху и снизу для наибольшего и наименьшего собственных чисел такой матрицы, то тогда $\alpha_k = \alpha - const = \frac{2(1-\varepsilon)}{M}$

21. Метод наискорейшего спуска, метод покоординатного спуска, сходимость градиентных методов.

Метод Коши (наискорейшего спуска)

$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \alpha_k f'(\bar{x}^{(k)})$; k=0,1,2….

α_k выбирается оптимальной величиной, обеспечивающей достижение min функции при движении в направлении $f'(\bar{x}^{(k)})$ из точки $\bar{x}^{(k)}$.

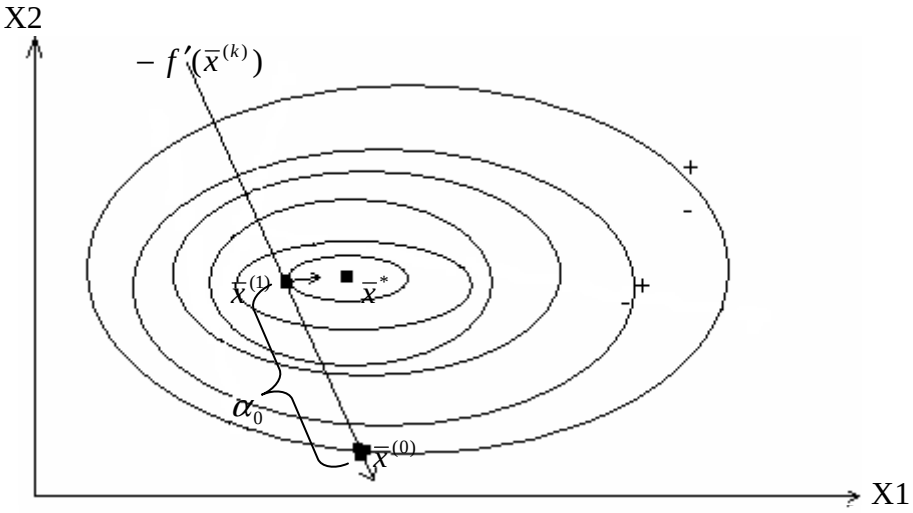
$$\alpha_k = \arg\min_{\alpha \geq 0} f\left(\bar{x}^{(k)} - \alpha f'(\bar{x}^{(k)})\right);$$

Направления $f'(\bar{x}^{(k)})$ и $f'(\bar{x}^{(k+1)})$ взаимно ортогональные:

$\varphi(\alpha) = f(\bar{x}^{(k)} - \alpha f'(\bar{x}^{(k)}))$ - направление ее min-ции;

$$\frac{d\varphi}{d\bar{x}} * \frac{d\bar{x}}{d\alpha} = -f'\left(\bar{x}^{(k)}\right) * f'\left(\bar{x}^{(k+1)}\right) = 0$$

Геометрическая интерпретация:



Градиентный метод покоординатного спуска.

Движение вдоль какой-либо оси на каждом шаге.

$$\begin{cases} \bar{x}^{(0)}; \bar{x}^{(1)} = \bar{x}^{(0)} - \alpha_0 \dfrac{\partial f(\bar{x}^{(0)})}{\partial x_1} * \bar{e}_1 \\ \bar{x}^{(1)}; \bar{x}^{(2)} = \bar{x}^{(1)} - \alpha_1 \dfrac{\partial f(\bar{x}^{(1)})}{\partial x_2} * \bar{e}_2 \\ \ldots \\ \bar{x}^{(kn+s-1)}; \bar{x}^{(kn+s)} = \bar{x}^{(kn+s-1)} - \alpha_{kn+s-1} \dfrac{\partial f(\bar{x}^{(kn+s-1)})}{\partial x_s} * \bar{e}_s \end{cases}$$

- спуск по всем направлениям;

Где $\bar{e}_i$ - столбец с нулями на всех местах кроме i-ого, на нем 1.

$k = 0,1,2....$

$s = \overline{1,n}$ k – число внешних итераций;

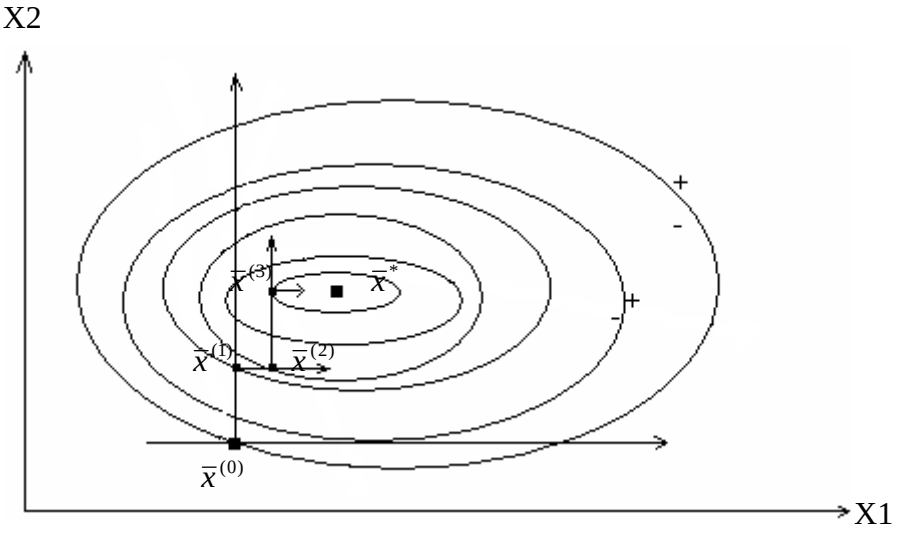
$$\begin{cases} x_j^{(kn+s)} = x_j^{(kn+s-1)};\ __\ j = \overline{1;n};\ __\ j \neq s; \\ x_j^{(kn+s)} = x_j^{(kn+s-1)} - \alpha_{kn+s-1} \dfrac{\partial f(\bar{x}^{kn+s-1})}{\partial x_s};\ __\ j = s; \end{cases}$$

Если α_{kn+s-1}=const, то имеем постоянный шаг;

Если α выбирается оптимальной длинной, то имеем покоординатный спуск, называемый методом Гаусса – Зейделя.

$$\alpha_{kn+s-1} = \arg_\min_{\alpha \geq 0}_ f\left(\bar{x}^{(kn+s-1)} - \alpha \frac{\partial f(\bar{x}^{(kn+s-1)})}{\partial x_s} * \bar{e}_s\right);$$

Геометрическая интерпретация:



<u>Теоремы о сходимости градиентных методов:</u>

1. Если f(x) ограниченно снизу и ее градиент удовлетворяет условию Липшица, т.е. выполняется соотношение: $\| f^{|}(\bar{y}) - f^{|}(x) \| \leq R \| \bar{y} - \bar{x} \|,\ \forall \bar{x}, \bar{y} \in E^n; y \neq x;$

$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \alpha_k f^{|}(\bar{x}^{(k)})$; k=0,1,2…. Где α_k удовлетворяет соотношению из градиентного метода с дроблением шага:

$$f\left(\bar{x}^{(k)} - \alpha_k f^{|}(\bar{x}^{(k)})\right) - f(\bar{x}^{(k)}) \leq -\xi\alpha_k \| f^{|}(\bar{x}^{(k)}) \|^2; \alpha_k > 0; __ 0 < \xi < 1;$$

То тогда справедливо, что $\| f^{|}(\bar{x}^{(k)}) \| \to 0, при_ k \to \infty, _ для_ \forall \bar{x}^{(0)}$;

Условие остановки: $\left| \dfrac{\partial f(\bar{x}^{(k)})}{\partial x} \right| \leq \xi$;

2. Если f(x) является непрерывно дважды деференцируемой функцией, для которой выполняется соотношение : $m \| \bar{y}^2 \| \leq \bar{y}^T f^{|}(\bar{x})\bar{y} \leq M \| \bar{x}^2 \|;\ \forall \bar{x}, \bar{y} \in E^n; y \neq x;$
то она называется сильно выпуклой функцией. Тогда для процесса

$\bar{x}^{(k+1)} \leq \bar{x}^{(k)} - \alpha_k f^{|}(\bar{x}^{(k)})$;

$\alpha_k = \arg_\min_{\alpha \geq 0}_ f\left(\bar{x}^{(k)} - \alpha f^{|}(\bar{x}^{(k)})\right)$, то для начальной точки $\bar{x}^{(0)}$ выполняется соотношение:

$\bar{x}^{(k)} \to \bar{x}^{(*)}; f(\bar{x}^{(k)}) \to f(\bar{x}^{(*)})$.

$$f\left(\bar{x}^{(k)}\right) - f\left(\bar{x}^{*}\right) \leq g^k \left(f\left(\bar{x}^{(0)}\right) - f\left(\bar{x}^{*}\right)\right)$$

Для скорости сходимости справедливы зависимости: $\| \bar{x}^{(k)} - \bar{x}^{*} \| \leq Cg^{k/2}$

$$g = (M - m)/(M + m);\ _ g \in [0;1]$$

$$C = const < \infty$$

M и m: равномерные по x оценки сверху и снизу, наибольшее и наименьшее собственные числа матрицы $f^{\|}(\bar{x})$. В качестве приближения можно взять собственные числа $f^{\|}(\bar{x}^{*})$.

22. Градиентный метод с масштабированием переменных.

(Для f вида $f = \sum_{j=1}^{n} \alpha_k x_j^2$, сепарабельных функций)

В случае когда g<<1 задача является хорошо обусловленной и решается достаточно быстро. А если M и m сильно отличаются друг от друга и g≈1, то задача является плохо обусловленной и требует большого числа итераций.

На примере:

$f(\bar{x}) = x_1^2 + 16x_2^2 \to \min \quad \bar{x}^{(0)} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \quad H = \begin{pmatrix} 2 & 0 \\ 0 & 32 \end{pmatrix}$ тогда выполняют замену переменных:

$y_i = \alpha_j^{1/2} x_j; \quad x_i = \alpha_j^{-1/2} y_j; \; j = \overline{1,n} \quad x_i = \mu_j y_j; \quad \mu_j = \alpha_j^{-1/2}$

$f(\bar{y}) = y_1^2 + y_2^2 \to \min$

$f^{|}(\bar{x}) = \begin{pmatrix} 2x_1 \\ 32x_2 \end{pmatrix} \quad f^{|}(\bar{x}^{(0)}) = \begin{pmatrix} 4 \\ 64 \end{pmatrix} \quad \bar{y}^{(0)} = \begin{pmatrix} 2 \\ 8 \end{pmatrix} \quad f^{|}(\bar{y}^{(0)}) = \begin{pmatrix} 4 \\ 16 \end{pmatrix}$

Для более сложных функций, компоненты μ_j пересчитываются на каждом шаге , по формуле $\mu_i = \left(\frac{\partial^2 f(\tilde{x})}{\partial x_i \partial x_i} \right)^{-1/2}; i = \overline{1,n}$, где $\tilde{x}$ - точка одномерного min-ма функции, при движении из $\bar{x}^{(k)}$ в направлении переменной x_i.

$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \alpha_k B_k f^{|}(\bar{x}^{(k)}) \quad B = \begin{pmatrix} \mu_1^2 & 0 \\ 0 & \mu_2^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 16 \end{pmatrix}$

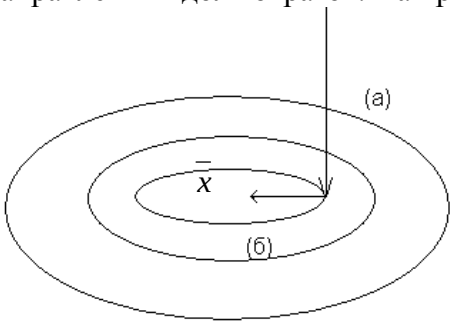
23. Эвристические схемы градиентных методов.

Эвристический метод с переключающимся алгоритмом.

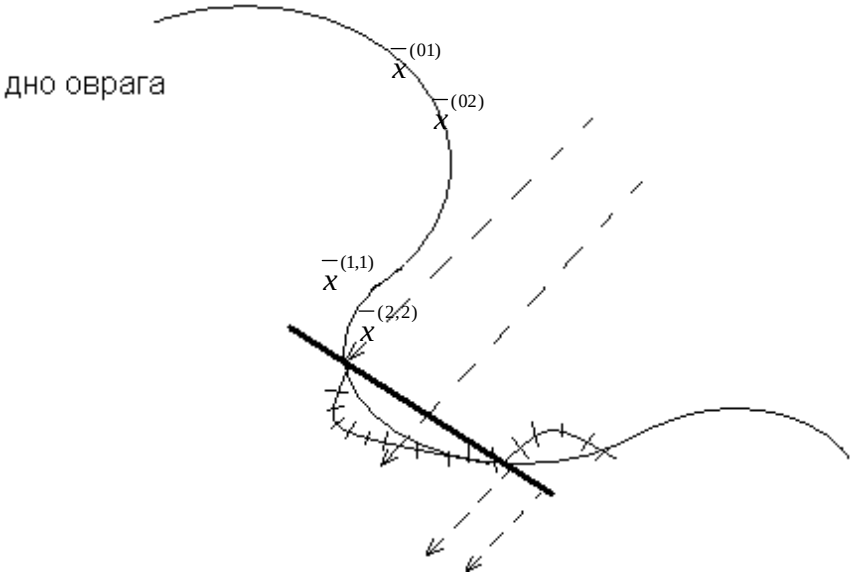
Применяется для функций со сложной топографий линий уровня. Состоит из 2 процедур:

1. Принимаем некоторое $\xi_1 << 1$ и затем принудительно полагаем $\frac{\partial f(\bar{x})}{\partial x_j} = 0$, x_j такие, что действительно выполняется соотношение $\left|\frac{\partial f(\bar{x})}{\partial x_j}\right| \le \xi_1$. Движение осуществляется ко дну оврага, в сторону сильно уменьшающихся значений функции.

2. Устанавливаем некоторое $\xi_2 >> 1$, принудительно полагаем $\frac{\partial f(\bar{x})}{\partial x_j} = 0$, для которых в действительности выполняется соотношение $\left|\frac{\partial f(\bar{x})}{\partial x_j}\right| \ge \xi_2$, в результате будим двигаться в направлении вдоль оврагов. На практике такие процедуры чередуют.



Метод Гельфанда (обладает линейной скоростью сходимостью)



24. Оптимизация многомерных функций методами второго порядка.

В методах второго порядка производиться квадратичная аппроксимация. Практически за один шаг обрабатывают любую квадратичную функцию. Для более сложных больше.

Метод Ньютона:

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - (f''(\bar{x}^{(k)}))^{-1} * f'(\bar{x}^{(k)})$$

Квадратичная скорость сходимости.

Метод Ньютона – Фафстона (модифицированный метод Ньютона с регулированием шага)

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \alpha_k \left(f''(\bar{x}^{(k)}) \right)^{-1} * f'(\bar{x}^{(k)});$$

$$\alpha_k = \arg_\min_ f\left[\bar{x}^{(k)} - \alpha \left(f''(\bar{x}^{(k)})^{-1} f'(\bar{x}^{(k)}) \right) \right];$$

$$f(\bar{x}^{(k+1)}) \le f(\bar{x}^{(k)});$$

Релаксационная последовательность.

25. Метод сопряженных градиентов.

Метод Флетчера-Ривса

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} + \alpha_k \bar{S}^{(k)}, k = 0,1,2...$$
$$\bar{S}^{(0)} = -f'(\bar{x}^{(0)})$$
$$\bar{S}^{(k)} = -f'(\bar{x}^{(k)}) + \beta_{k-1}\bar{S}^{(k-1)} \; ? k = 1,2,...$$

Направления S были взаимносопряженными.

$$\bar{S}^{(k)}, \bar{S}^{(k-1)}; f(\bar{x}) = a + (\bar{b}\bar{x}) + \frac{1}{2}(\bar{x}, H\bar{x}) \to \min$$

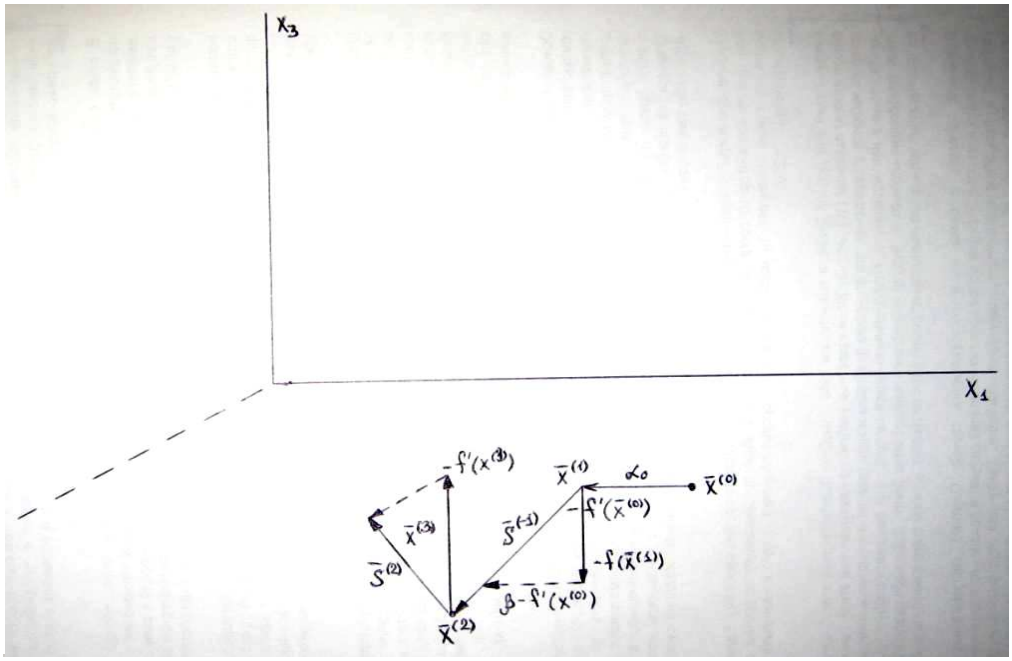
$$(\bar{S}^{(k)}; H\bar{S}^{(k-1)}) = 0; = -(f'(\bar{x}^{(k)}), H\bar{S}^{(k-1)}) + \beta_{k-1}(\bar{S}^{(k-1)}, H\bar{S}^{(k-1)});$$
$$\beta_{k-1} = \frac{(f'(\bar{x}^{(k)}, H\bar{S}^{(k-1)})}{(\bar{S}^{(k-1)}, H\bar{S}^{(k-1)})}$$

Длину шага выбирают оптимальной и для любой квадратичной функции:

$$\alpha_k = \frac{f'(\bar{x}^{(k)}, \bar{S}^{(k)})}{(\bar{S}^{(k)}, H\bar{S}^{(k)})};$$

Для более сложных функций:

$$\alpha_k = \arg\min_{\alpha \geq 0} f(\bar{x}^{(k)} + \alpha\bar{S}^{(k)});$$



$$\bar{S}^{(k-1)}, f'^{(k)}, \beta_{k-1}$$
$$\bar{x}^{(k+1)} = \bar{x}^{(k)} + \alpha_k \bar{S}^{(k)} H$$
$$H\bar{x}^{(k+1)} = H\bar{x}^{(k)} + \alpha_k H\bar{S}^{(k)}$$
$$\underbrace{\bar{b} + H\bar{x}^{(k+1)}}_{f'(\bar{x}^{(k+1)})} = \underbrace{\bar{b} + H\bar{x}^{(k)}}_{f'(\bar{x}^{(k)}) \to f(\bar{x}) = a + (\bar{x}, \bar{b}) + \frac{1}{2}(\bar{x}, H\bar{x})} + \alpha_k H\bar{S}^{(k)}$$

$$\alpha_k H\bar{S}^{(k)} = f(\bar{x}^{(k+1)}) - f(\bar{x}^{(k)}),$$
$$\alpha_{k-1} H\bar{S}^{(k-1)} = f'(\bar{x}^{(k)}) - f'(\bar{x}^{(k-1)}),$$
$$\beta_{k-1} = \frac{(f'(\bar{x}^{(k)}), H\bar{S}^{(k-1)})}{(\bar{S}^{(k-1)}, H\bar{S}^{(k-1)})} = \frac{(f'(\bar{x}^{(k)}), f'(\bar{x}^{(k)}) - (f'(\bar{x}^{(k)}), f'(\bar{x}^{(k-1)}))}{(\bar{S}^{(k-1)}, \bar{f}(\bar{x}^{(k)}) - (\bar{S}^{(k-1)}, f'(\bar{x}^{(k-1)}))} =$$
$$= \frac{(f'(\bar{x}^{(k)}), f'(\bar{x}^{(k)}))}{-(-f'(\bar{x}^{(k-1)}), f'(\bar{x}^{(k-1)}) - \beta_{k-2}(\bar{S}^{(k-1)}, \bar{S}^{(k-2)})} = \frac{(f'(\bar{x}^{(k)}), f'(\bar{x}^{(k)}))}{(f'(\bar{x}^{(k-1)}), f'(\bar{x}^{(k-1)}))}$$

Алгоритм:

1.Задаем $\bar{x}^{(0)}, \bar{S}^{(0)} = -f'(\bar{x}^{(0)})$
2.k-ый шаг. Попадаем первый раз с параметрами . Вычисляем оптимальную длину шага.
$\alpha_k = \arg\min_{\lambda \geq 0} f(\bar{x}^{(k)} + \alpha\bar{S}^{(k)})$;
$\bar{x}^{(k+1)} = \bar{x}^{(k)} + \alpha_k \bar{S}^{(k)}$
3.Осуществляем проверку.
Вычисляем:
$f(\bar{x}^{(k+1)})$
$f'(\bar{x}^{(k+1)})$

Проводим проверку на остановку алгоритма. Если $\| f'(\bar{x}^{(k+1)}) \| \leq \varepsilon \rightarrow$ останавливаем алгоритм.
$\bar{x}^{(k+1)} - приближение_к_\bar{x}^*$
4.Вычисляем коэффициент

$$\beta_k = \frac{(f'(\bar{x}^{(k+1)}), f'(\bar{x}^{(k+1)}))}{(f'(\bar{x}^{(k)}), f'(\bar{x}^{(k)}))}$$

Вычисляем $\bar{S}^{(k+1)} = -f'(\bar{x}^{(k+1)}) + \beta_k \bar{S}^{(k)}$ и переходим к п.2 алгоритма.

Для любой квадратичной функции дает гарантированное решение за n шагов(суперлинейная сходимость). Алгоритм первого порядка, т.к. нет вторых производных.

Для более сложных функций после каждой n итерации работу алгоритма нужно повторять в кол-ве $\bar{x}^{(0)}$ для нового повторения брать последнюю точку из пред. работы алгоритма.

26. Теорема Куна-Таккера, доказательство достаточности.

(1) $f(\bar{x}) \to \min$

(2) $\tilde{g}_i(\bar{x}) \le 0, i = \overline{1,m}$

(1),(2) задача выпуклого программирования, т.е. $f(\bar{x})_u_\tilde{g}(x)$ явл. выпуклыми функциями.

Условие регулярности Слейтера

$\bar{x} \in X$ (является внутренней точкой X) и для неё выполняется ограничение:

$\tilde{g}_i(\bar{x}) \le 0, i = \overline{1,m}$

n-мерный объем (X)

$f(\bar{x}), \tilde{g}_i(\bar{x}), i = \overline{1,m}$ - являются выпуклыми функциями.

Функция Лагранжа:

(3) $F(\bar{x}, \bar{\lambda}) = f(\bar{x}) \ne \sum_{i=1}^{n} \lambda_i \tilde{g}_i(\bar{x}),$

Теорема

Для того чтобы $\bar{x}^*$ являлся оптимальным вектором задачи (1),(2) необходимо и достаточно:

$\exists \bar{\lambda}^* \ge 0$,для которой $\bar{\lambda} \ge 0$ выполняется след. соотношение

(4) $f(\bar{x}^*, \bar{\lambda}) \le f(\bar{x}^*, \bar{\lambda}^*) \le f(\bar{x}, \bar{\lambda}^*)$,т.е. $f(\bar{x}^*, \bar{\lambda}^*)$ -седловая точка функции Логранжа (3)

$\Rightarrow$ в ней достигается min $f(\bar{x}, \bar{\lambda})$ по $\bar{x}$ и max по среди $\forall_\bar{\lambda} \ge 0$

Док-во: Достаточность

Считается, что (4) выполняется для $\forall_\bar{\lambda} \ge 0$. И покажем, что $\bar{x}^*$ является решением з.(1)-(2).

$$\Delta \le f(\bar{x}^*) + \sum_i \lambda_i^* \tilde{g}_i(\bar{x}^*) \le f(\bar{x}) + \sum_i \lambda_i^* \tilde{g}_i(\bar{x})$$

$$\Delta = f(\bar{x}^*) + \sum_i \lambda \tilde{g}_i(\bar{x}^*), \forall \lambda \ge 0 \quad \text{___(5)}$$

$\Rightarrow$ *выполняеться* $_u_\lambda_i \rangle \lambda_i^* \Rightarrow \tilde{g}_i(\bar{x}^*) \le 0, i = \overline{1,m}$

$\Rightarrow (\bullet) \bar{x}^* \in X$

Является допустимым решением задачи (1)-(2).

Левое (Δ) выполняется и при $\lambda_i = 0$

$\Rightarrow \sum_i A_i^* \tilde{g}_i(\bar{x}^*) = 0$

С учетом этого $\prec (\Pi)$: $f(\bar{x}) + \sum_i \underbrace{\underbrace{\lambda_i^*}_{\ge 0} \underbrace{\tilde{g}_i}_{\le 0}}_{\le 0}(\bar{x})$

$f(\bar{x}^*) \le f(\bar{x})$ для $\forall \bar{x} \in X \Rightarrow \bar{x}^*$ точка глобального min, т.е. оптим-я.

27. Теорема Куна-Таккера, доказательство необходимости.

(1) $f(\bar{x}) \to \min$

(2) $\tilde{g}_i(\bar{x}) \le 0, i = \overline{1,m}$

(1),(2) задача выпуклого программирования, т.е. $f(\bar{x})_u_\tilde{g}(x)$ явл. выпуклыми функциями.

Условие регулярности Слейтера

$\bar{x} \in X$ (является внутренней точкой X) и для неё выполняется ограничение:

$\tilde{g}_i(\bar{x}) \langle 0, i = \overline{1,m}$

n-мерный объем (X)

$f(\bar{x}), \tilde{g}_i(\bar{x}), i = \overline{1,m}$ - являются выпуклыми функциями.

Функция Лагранжа:

(3) $F(\bar{x}, \bar{\lambda}) = f(\bar{x}) \ne \sum_{i=1}^{n} \lambda_i \tilde{g}_i(\bar{x}),$

Теорема

Для того чтобы $\bar{x}^*$ являлся оптимальным вектором задачи (1),(2) необходимо и достаточно:

$\exists \bar{\lambda}^* \ge 0$,для которой $\bar{\lambda} \ge 0$ выполняется след. соотношение

(4) $f(\bar{x}^*, \bar{\lambda}) \le f(\bar{x}^*, \bar{\lambda}^*) \le f(\bar{x}, \bar{\lambda}^*)$,т.е. $f(\bar{x}^*, \bar{\lambda}^*)$ -седловая точка функции Логранжа (3)

$\Rightarrow$ в ней достигается min $f(\bar{x}, \bar{\lambda})$ по $\bar{x}$ и max по среди $\forall_\bar{\lambda} \ge 0$

Док-во: Необходимость

Считаем, что $\bar{x}^*$ является оптимальным решением задачи (1)-(2),необходимо найти: $\bar{\lambda}^* \ge 0$, при котором выполняется (4)

Построим 2 вспомогательных множества точек:

$$M^{(1)} = \left\{ \bar{y}^{(1)} \right\} \begin{matrix} y_0 \ge f'(\bar{x}) \\ y_1 \ge \tilde{g}_1(\bar{x}) \\ i..... \\ y_m \ge \tilde{g}_m(\bar{x}) \end{matrix}, \bar{x} \in E^n \ \}$$

$$M^{(2)} = \left\{ \bar{y}^{(2)} \right\} \begin{matrix} y_0 \langle f(\bar{x}^*) \\ y_1 \langle 0 \\ ..... \\ y_m \langle 0 \end{matrix} \ \};$$

$M^{(1)}, M^{(2)}$ выпуклые множества, неперсек-ся $\Rightarrow$ для них м. построить гипер пл-ть вида:

$(\bar{V}, \bar{y}) = C, C = const$

Разделяющую эти множества, для которой применительно к $M^{(1)}, M^{(2)}$:

(6) $V^T \cdot \bar{y}^{(1)} \geq V^T \cdot \bar{y}^{(2)}$

$\prec$(6) применительно к $\bar{V} \neq 0$; т.к. компонент $\bar{y}^{(2)}$ не ограничено снизу и м. принимать неограниченно большие значения $\Rightarrow \bar{V} \geq 0$

|| (6) справедливо и для нестрогого разделения $M^{(1)}, M^{(2)}$

Пусть $\bar{y}^{(1)} = \begin{pmatrix} f(\bar{x}) \\ \tilde{g}_1(\bar{x}) \\ ..... \\ \tilde{g}_m(\bar{x}) \end{pmatrix}, \bar{y}^{(2)} = \begin{pmatrix} f(\bar{x}^*) \\ 0 \\ ..... \\ 0 \end{pmatrix}$

Подставим в (6) и получим:

$V_0 f(\bar{x}) + \sum_{i=1}^{m} V_i \tilde{g}_i(\bar{x}) \geq V_0 f(\bar{x}^*), \forall x \in X$ (7)

$\Rightarrow V_0 \neq 0, очевидно.$

Допустим, что это не так, т.е. $V_0 = 0$, тогда из (7) $\Rightarrow$ что

$\sum_{i=1}^{m} V_i \tilde{g}_i(\bar{x}) \geq 0, \forall x \in X$,но этого не может быть, т.к. $\bar{V} \geq 0, \bar{V} \neq 0$, т.е. хотя бы один компонент $V_i \rangle 0$, но при этом в X есть хотя б одна $\bar{x}$, в котором по условию Слейтера:

$\tilde{g}_i(\bar{x}) \langle 0, i = \overline{1,m}$

Значит наше предположение не верно $\Rightarrow V_o \neq 0$,а именно $V_o \rangle 0$

$\lambda_i^* = \frac{V_i}{V_o}, i = \overline{1,m}$

Тогда (7): V_0

(8) $f(\bar{x}) + \sum_{i=1}^{m} \lambda_i^* \tilde{g}_i(\bar{x}) \geq f(\bar{x}^*), \forall x \in X$

А значит оно справедливо и для $(\bullet)\bar{x}^*$, т.е.

(9) $f(\bar{x}^*) + \sum \lambda_i^* \tilde{g}_i(\bar{x}^*) \geq f(\bar{x}^*); \Rightarrow \sum_{i=1}^{m} \underbrace{\lambda_i^*}_{\leq 0} \underbrace{\tilde{g}_i(\bar{x}^*)}_{\leq 0} \geq 0; \Rightarrow$

(10) $\sum_{i=1}^{m} \lambda_i^* \tilde{g}_i(\bar{x}^*) = 0; \Rightarrow$

(11) $f(\bar{x}^*, \bar{\lambda}^*) = f(\bar{x}^*)$, выполняется частично

$\prec$(8) с учетом (11) можем записать

$f(\bar{x}, \bar{\lambda}^*) \geq f(\bar{x}^*)$ (12)

Разумеется, справедлива запись

$(\Delta): f(\bar{x}^*) \geq f(\bar{x}^*) + \sum_{i=1}^{m} \lambda_i \tilde{g}_i(\bar{x}^*)$ (13)

Запишем (13) с учетом (11):

(14) $f(\bar{x}^*, \bar{\lambda}^*) \geq f(\bar{x}^*, \bar{\lambda})$

Объединим (12) и (14):

$f(\bar{x}^*, \bar{\lambda}) \leq f(\bar{x}^*, \bar{\lambda}^*) f(\bar{x}, \bar{\lambda}^*)$

28. Развитие и обобщение метода Лагранжа, общая теорема математического программирования.

Развитие метода

$$\begin{cases} f(\bar{x}) \to \min & (1) \\ \tilde{g}_i(\bar{x}) = 0, \forall i = \overline{1,m} & (2) \\ x_j \geq 0,\ j = \overline{1,n} & (3) \end{cases}$$

$f(\bar{x}), \tilde{g}_i(\bar{x})$ являются выпуклыми непрерывно дифференцируемыми функциями.

Найдем набор необходимых условий для того, чтобы $(\bar{x}^*)$ являлась оптимальным планом задания $(1) \div (3)$, а $(\bar{x}^*, \bar{\lambda}^*)$ седловой $(\cdot)$ функции Лагранжа.

$\prec \forall компонента_\bar{x}^*$

1) $x_j^* \rangle 0$ к любому такому компоненту мы можем применять малые вариации увеличения и уменьшения применительно к $f(\bar{x}^*, \bar{\lambda}^*)$

$$\frac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial x_j} = 0 \text{ (необходимое условие)}$$

2) $x_j^* = 0$ можем только увеличить значение при анализе функции Лагранжа. Тогда необходимое условие:

$$\frac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial x_j} \geq 0$$

Окончательно весь набор необходимых условий:

(4) $$\frac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial x_j} = \frac{\partial f(\bar{x}^*)}{\partial x_j} + \sum_{i=1}^{m} \lambda_i^* \frac{\partial \tilde{g}_i(\bar{x}^*)}{\partial x_j} \geq 0,\ j = \overline{1,n}$$

(5) $$\frac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial x_j} = 0,\ j = \overline{1,n}$$

(6) $$x_j \geq 0,\ j = \overline{1,n}$$

(7) $$\frac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial \lambda_j} = \tilde{g}_i(\bar{x}^*) = 0, i = \overline{1,m}$$

Обобщение метода

Задача вида:

Выпуклые $$\begin{cases} f(\bar{x}) \to \min \\ \tilde{g}_i(\bar{x}) \leq 0,\ j = 1,m \\ x_j \geq 0,\ j = 1,m \end{cases}$$

Найдем набор необходимых условий:

Добавим $y_i \geq 0, i = \overline{1,m}$, тогда получим (1), (2), (3) из «Развития метода» и применим к зад. Условия $(4) \div (7)$ из «Развития метода».

$$f(\bar{x}, \bar{\lambda}, \bar{y}) = f(\bar{x}) + \sum_{i=1}^{m} \lambda_i \tilde{g}_i(\bar{x}) + \sum_{i=1}^{m} \lambda_i y_i \to \min$$

(1) $$\frac{\partial f(\bar{x}^*, \bar{\lambda}^*, \bar{y}^*)}{\partial x_j} \geq 0,\ j = \overline{1,n}$$

(2) $x_j^* \cdot \frac{\partial f(\bar{x}^*, \bar{\lambda}^*, \bar{y}^*)}{\partial x_j} = 0,\ j = \overline{1,n}$

(3) $x_j^* \geq 0,\ j = \overline{1,n}$

(4) $\frac{\partial f(\bar{x}^*, \bar{\lambda}^*, \bar{y}^*)}{\partial y_j} = \lambda_i^* \geq 0, i = \overline{1,m}$

(5) $y_i^* \cdot \frac{\partial f(\bar{x}^*, \bar{\lambda}^*, \bar{y}^*)}{\partial y_j} = 0, i = \overline{1,m}$

(6) $y_i^* \geq 0, i = \overline{1,m}$

(7) $\frac{\partial f(\bar{x}^*, \bar{\lambda}^*, \bar{y}^*)}{\partial \lambda_j} = \tilde{g}_i(\bar{x}^*) + y_i^* = 0$

Из (4) следует, что $\lambda_i^* \geq 0$ для $\forall i = \overline{1,m}$

Из (5) следует, что в (.) $(\bar{x}^*, \bar{\lambda}^*)$ расширенная функция.

Из (5) следует, что $\bar{y}^*, \lambda^* = 0, i = \overline{1,m};$

$f(\bar{x}^*, \bar{\lambda}^*, \bar{y}^*) = f(\bar{x}^*, \bar{\lambda}^*)$ числено

В (7) заметим $\tilde{g}_i(\bar{x}^*)$ выразим из предыдущего пункта:

$$\frac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial \lambda_i} = \tilde{g}_i(\bar{x}^*), i = \overline{1,m};$$

$$y_i^* \geq 0, i = \overline{1,m}$$

Таким образом

$\frac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial \lambda_i} \leq 0, i = \overline{1,m}$ (9)

Тогда (9) с учетом (5) дает

$\lambda_i^* \cdot \frac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial \lambda_i} = 0, i = \overline{1,m}$ (10)

Окончательно объединим (1),(2),(3),(8),(9),(10) получаем набор окончательных условий для седловой точки:

$$\begin{cases} \frac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial \bar{x}} \geq 0 \\ \bar{x}^* \cdot \frac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial \bar{x}} = 0 \\ \bar{x}^* \geq 0 \\ \frac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial \bar{\lambda}} \leq 0 \\ \bar{\lambda}^* \cdot \frac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial \bar{\lambda}} = 0 \\ \bar{\lambda}^* \geq 0 \end{cases}$$

ОТМП

Для того, чтобы точка $\bar{x}^* \geq 0$ являлась (.) глобального экстремума функции на множестве ограничений (2),(3), т.е. на выпуклом множестве, необходимо, чтобы $\exists \bar{\lambda}^* \geq 0$, при котором выполняются необходимые условия:

1) Для $[f(\bar{x}) \to \min]$

$$\begin{cases} f'_{\bar{x}}(\bar{x}^*, \bar{\lambda}^*) \geq 0 \\ (\bar{x}^*, f'_{\bar{x}}(\bar{x}^*, \bar{\lambda}^*)) = 0, \bar{x}^* \geq 0 \\ f'_{\bar{\lambda}}(\bar{x}^*, \bar{\lambda}^*) \leq 0 \\ (\bar{\lambda}^*, f'_{\bar{\lambda}}(\bar{x}^*, \bar{\lambda}^*)) = 0, \bar{\lambda}^* \geq 0 \end{cases}$$

2) 2) Для $[f(\bar{x}) \to \max]$

$$\begin{cases} f'_{\bar{x}}(\bar{x}^*, \bar{\lambda}^*) \leq 0 \\ (\bar{x}^*, f'_{\bar{x}}(\bar{x}^*, \bar{\lambda}^*)) = 0, \bar{x}^* \geq 0 \\ f'_{\bar{\lambda}}(\bar{x}^*, \bar{\lambda}^*) \geq 0 \\ (\bar{\lambda}^*, f'_{\bar{\lambda}}(\bar{x}^*, \bar{\lambda}^*)) = 0, \bar{\lambda}^* \geq 0 \end{cases}$$

|| Если функция и множество строго выпуклые эти условия достаточные.

29. Общая теорема математического программирования, условия оптимальности для задач квадратичного программирования.

ОТМП

Для того, чтобы точка $\bar{x}^* \geq 0$ являлась (.) глобального экстремума функции на множестве ограничений (2),(3), т.е. на выпуклом множестве, необходимо, чтобы $\exists \bar{\lambda}^* \geq 0$, при котором выполняются необходимые условия:

1) Для $[f(\bar{x}) \to \min]$

$$\begin{cases} f'_{\bar{x}}(\bar{x}^*, \bar{\lambda}^*) \geq 0 \\ (\bar{x}^*, f'_{\bar{x}}(\bar{x}^*, \bar{\lambda}^*)) = 0, \bar{x}^* \geq 0 \\ f'_{\bar{\lambda}}(\bar{x}^*, \bar{\lambda}^*) \leq 0 \\ (\bar{\lambda}^*, f'_{\bar{\lambda}}(\bar{x}^*, \bar{\lambda}^*)) = 0, \bar{\lambda}^* \geq 0 \end{cases}$$

2) 2) Для $[f(\bar{x}) \to \max]$

$$\begin{cases} f'_{\bar{x}}(\bar{x}^*, \bar{\lambda}^*) \leq 0 \\ (\bar{x}^*, f'_{\bar{x}}(\bar{x}^*, \bar{\lambda}^*)) = 0, \bar{x}^* \geq 0 \\ f'_{\bar{\lambda}}(\bar{x}^*, \bar{\lambda}^*) \geq 0 \\ (\bar{\lambda}^*, f'_{\bar{\lambda}}(\bar{x}^*, \bar{\lambda}^*)) = 0, \bar{\lambda}^* \geq 0 \end{cases}$$

|| Если функция и множество строго выпуклые эти условия достаточные.

Условия для задач квадратичного программирования.

$$\begin{cases} f(\bar{x}) = \sum_{j=1}^{m} c_j x_j + (d_{11}x_1^2 + d_{22}x_2^2 + \ldots + d_{nn}x_n^2 + 2d_{12}x_1x_2 + 2d_{13}x_1x_3 + \ldots + 2d_{n-1,n}x_{n-1}x_n) \to \max \\ \sum_{j=1}^{n} a_{ij}x_j = b_i, i = \overline{1,m} \\ x_j \geq 0, j = \overline{1,n} \end{cases}$$

$f(\bar{x})$ строго вогнута.

Тогда, на выпуклом множестве для такой $f(x)$ **ОТМП** является достаточным условием:

$$\bar{c} = \begin{pmatrix} c_1 \\ c_2 \\ .. \\ c_n \end{pmatrix}, \bar{b} = \begin{pmatrix} b_1 \\ b_2 \\ .. \\ b_m \end{pmatrix}, \bar{x} = \begin{pmatrix} x_1 \\ x_2 \\ .. \\ x_n \end{pmatrix}, A_{m \times n}$$

$$D = \begin{pmatrix} d_{11}\ d_{12}\ \ldots\ d_{1n} \\ d_{21}\ d_{22}\ \ldots\ d_{2n} \\ \ldots\ldots\ldots\ldots\ldots \\ d_{n1}\ d_{n2}\ \ldots\ d_{nn} \end{pmatrix}$$

Запишем з. в векторном виде:

$$\begin{cases} f(\bar{x}) = \bar{c}^T \bar{x} + \bar{x}^T D\bar{x} \to \max \\ A\bar{x} = \bar{b} \\ \bar{x} \geq 0 \end{cases}$$

Функция Лагранжа для $f(x)$:

$f(\bar{x}, \bar{\lambda}) = \bar{c}^T \bar{x} + \bar{x}^T Dx + \bar{\lambda}^T (\bar{b} - A\bar{x})$

Запишем $f(\bar{x})$ с использованием матрицы Гессе:

$$f(\bar{x}) = \bar{c}^T \bar{x} + \frac{1}{2} \bar{x}^T H\bar{x}$$

Применим ОТМП к функции Лагранжа:

(1) $\dfrac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial x} = \bar{c} + 2D\bar{x}^* - A^T \bar{\lambda}^* \leq 0$

(2) $\bar{x}^* \cdot \dfrac{\partial f(\bar{x}^*, \bar{\lambda}^*)}{\partial x} = 0$

(3) $\bar{x}^* \geq 0$

(4) $A\bar{x}^* = \bar{b}$

Добавляем в левую часть (1) дополнит. перемен-е $\bar{V} \geq 0$, тогда получим:

(5) $\bar{c} + 2D\bar{x}^* - A^T \bar{\lambda}^* + \bar{V}^* = 0$

(6) $\bar{V}^* = -\bar{c} - 2D\bar{x}^* + A^T \bar{\lambda}^* \geq 0$

Тогда, с учетом (5),(6) набор необходимых и достаточных условий для исходной задачи квадратичного программирования:

$$\begin{cases} 2D\bar{x}^* - A^T \bar{\lambda}^* + \bar{V} = -\bar{c} \\ \bar{x}^{*T} \cdot \bar{V} = 0 \\ A\bar{x}^* = \bar{b} \\ \bar{x}^* \geq 0, \bar{V} \geq 0 \end{cases}$$

Если для такой задачи существуют вектора $\bar{x} \rangle 0, \bar{\lambda} \rangle 0, \bar{V} \rangle 0$, то исходная задача разрешима.

30. Метод Била.

$$\begin{cases} f(\bar{x}) = \bar{c}^T \bar{x} + \bar{x}^T Dx \to \max \\ A\bar{x} = \bar{b} \\ \bar{x} \geq 0 \end{cases}$$ квадратичная форма

X - выпуклое множество.
$f(\bar{x})$ - вогнутая функция.

Алгоритм начинает работу от 1ДБР исх. зад.
Пусть мы нашли такое 1ДБР, т.е. т БП выразили через остальные $x_{m+1}, x_{m+2}, ..., x_n$ - НБП и получили:

$$\begin{cases} БП \begin{cases} x_1 = ... \\ x_2 = ... \\ ..... \\ x_m = ... \end{cases} \\ x_j \geq 0,\ j = 1, n \\ f(\bar{x}) = (c_{0,0} + c_{0,m+1}x_{m+1} + c_{0,m+2}x_{m+2} + ... + c_{0,n}x_n) \cdot 1 + (c_{m+1,0} + c_{m+1,m+1}x_{m+1} + c_{m+1}x_{m+2} + ... + \\ + c_{m+1,n}x_n)x_{m+1} + ... + (c_{n,0} + c_{n,m+1}x_{m+1} + ... + c_{n,n}x_n)x_n \to \max \end{cases}$$

(!) $\frac{\partial f(\bar{x})}{\partial x_i} \leq 0, i = \overline{1, m}$ условие оптимальности

$\frac{1}{2}\frac{\partial f(\bar{x})}{\partial x_h} = c_{h,0}$; $f(\bar{x}) = c_{0,0} - должныполучить$

$\leftarrow c_{ij} = c_{ji}$

Допустим, что $c_{0,m+1} \rangle 0$, тогда если мы увеличим x_{m+1} и вводить V в очередной базис, то $f(\bar{x})$ будет возрастать.

// В обычном симплекс-алгоритме мы бы V увел., пока к-л БП обратилось в «не существует», но здесь возможно что

$\frac{1}{2}\frac{\partial f(\bar{x})}{\partial x_{m+1}}$ обратиться в «не существует» раньше какой либо БП и тогда при продолжении увеличения скобка с x_{m+1} станет принимать отрицательные значения $\Rightarrow$ $f(x)$ будет убывать, что недопустимо.

При исполнении симплекс-алгоритма, может возникнуть ситуация, что:

$$x_h, \frac{1}{2}\frac{\partial f(\bar{x})}{\partial x_h} = 0$$

В этом случае вводим новую переменную, неограниченную знаком:

$$u_1 = \frac{1}{2}\frac{\partial f(\bar{x})}{\partial x_h}$$

$x_h = ...$

В начале очередной итерации НБП будут все прежние, за исключением x_h и +переменная u_1; а БП будут прежние + x_h

Поскольку u_1 неограниченна по знаку, то если:

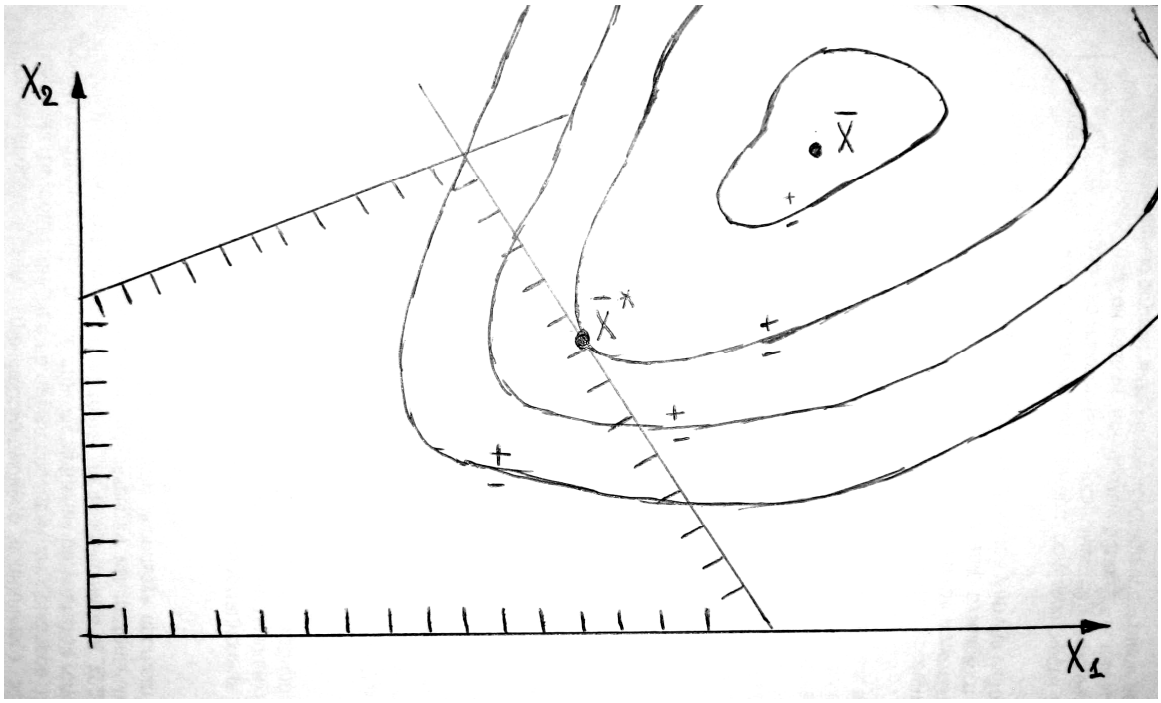
1) $\frac{\partial f(\bar{x})}{\partial u_1} \rangle 0$ то значение u_1 надо увеличить

При этом $f(\bar{x})$ будет возрастать. u_1 увеличивать необходимо до тех пор, пока $\frac{\partial f(\bar{x})}{\partial u_1} = 0$

2) $\frac{\partial f(\bar{x})}{\partial u_1} \langle 0$ то значение u_1 надо уменьшать, пока $\frac{\partial f(\bar{x})}{\partial u_1} = 0$. При этом значение $f(\bar{x})$ будет также возрастать.

Если же все производные $\left(\frac{\partial f}{\partial u_i}\right)$ по всем u_i равны «не существует», то проводим анализ $\frac{\partial f(\bar{x})}{\partial x_h}$ (по НБП)

Если все $\frac{\partial f(\bar{x})}{\partial x_h} \le 0$, то мы получим max значение $f(\bar{x})$:



$$f(\bar{x}) = 10x_1 - x_1^2 + 2x_1x_2 - 2x_2^2 \to \max$$
$$x_1 + 2x_2 \le 10$$
$$x_1 + x_2 \le 6$$
$$x_1, x_2 \ge 0$$

$$
\text{1ДБР}\begin{cases}
x_3 = 10 - x_1 - 2x_2 \quad (a) \\
x_4 = 6 - x_1 - x_2 \quad (b) \\
x_j \geq 0,\ j = \overline{1,4} \\
f(\bar{x}) = 5x_1 + \overbrace{(5 - x_1 + x_2)}^{\substack{x_{1\max}=5(\text{при больших } x_1 \text{ начнет} \\ \text{убывать функция цели})}} \quad x_1 + (x_1 - 2x_2)x_2 \to \max \\
x_{1\max} = 10\ (no(a))\ при\ x_2 = 0 \\
x_{1\max} = 6\ (no(b))\ при\ x_2 = 0
\end{cases}
$$

Вводим u_1:

$$\frac{1}{2}\frac{\partial f}{\partial x_1} = (5 - x_1 + x_2) = u_1$$

В 1ДБР мы находим max значение для x_1 и вводим новую переменную.

Новое ДБР:

$$
\text{2ДБР}\begin{cases}
x_3 = 5 + u_1 - 3x_2 \\
x_4 = 5 - u_1 - 2x_2 \\
x_1 = 5 - u_1 + x_2 \\
x_j \geq 0,\ j = \overline{1,4} \\
f(x) = (25 + 5x_2)\cdot 1 + (-u_1)u_1 + (5 - x_2)x_2 \to \max
\end{cases}
$$

Начальная таблица по 1ДБР:

Начальная таблица по 1ДБР

БП	1	X_1	X_2
X_3	10	-1	-2
X_4	6	-1	-1
X_1	0	1	0
X_2	0	0	1
1	0	5	0
X_1	5	-1	1
X_2	0	1	2

Промежуточная таблица

НБП / БП	1	U_1	X_2
X_3	5	1	-3
X_4	1	1	-2
X_1	5	-1	1
X_2	0	0	1
1	25	-5	5
X_1	0	1	0
X_2	5	-1	-1

Окончательная таблица (начальная для след. итерации)

НБП / БП	1	U_1	X_2
X_3	5	1	-3
X_4	1	1	-2
X_1	5	-1	1
X_2	0	0	1
1	25	0	5
X_1	0	-1	0
X_2	5	0	-1

Алгоритм

Выбор разрешающего столбца.

Если в Ioй строке функции цели все(коэфф. при переменных u_i)=0 и все(коэфф. при x_j)=0, то текущее решение является оптимальным.

Иначе, в качестве разрешающего столбца выбираем любой столбец u_i, в которой в l-ой строке функции ненулевой коэфф.
Если все(коэфф. u_i)=0 в l-ой строке функции или таких столбцов нет, то в качестве разрешающего столбца выбираем любой с переменной x_j с положит. коэфф. в l-ой строке функции цели.

2)Определение разрешающей строки.

Найдем отношение для нижней части таблицы: $\left|\frac{c_{0j}}{cjj}\right|$ причем это вычисление выполняется лишь тогда, когда $cjj\langle 0$;

cjj -это элемент, стоящий на пересечении разрешающего столбца и главной диагонали нижней части таблицы.
Для верхней части таблицы вычисляем отношение свободных членов к модулям коэфф. из разрешающего столбца.(рассмотрим только для тех строк, где эти коэфф. противоположны по знаку с c_{0j}). Сравниваем все найденные отношения. Строка, дающая min из этих отношений, берется в качестве первой разрешающей строки.
3.Заполнение промежуточной таблицы.

В разрешающем столбце в промежуточной таблице занесена переменная x_j из разрешающей строки начальной таблицы, если эта разрешающая строка находится в верхней части таблицы. А если в нижней(начальная таблица) части, то в разрешающий столбец промежуточной таблицы записываем очередную промежуточную переменную u_i.

Все элемента разрешающего столбца делим на генеральный элемент и записываем в промежуточную таблицу. Остальные элементы разрешающей строки делаем равными «не существует». Прочие элементы пересчитываем по обычному правилу прямоугольника.

Строка у нижней части таблицы, которая пересекается с разрешающим столбцом на главной диагонали нижней части таблицы называется 1ой разрешающей строкой.

Если 1ая разрешающая строка находится в нижней части таблицы, то 1ая и 2ая разрешающие строки совпадают.

4) Заполнение конечной таблицы.
Верхняя часть полностью совпадает с верхней частью промежуточной таблицы. На место заглавное переменной во 2ой разрешающей строке конечной таблицы записывается имя переменной из разрешающего столбца промежуточной таблицы. Делим 2ую разрешающую строку промежуточной таблицы на генеральный элемент из начальной таблицы, получаем соответствующую строка в конечной таблице.
Прочая h-ая строка получается, если из h-ой строки промежуточной таблицы вычесть 2ую разрешающую строку конечной таблицы умноженную на h-ый элемент из l-ой разрешающей строки начальной таблицы.

```
_  25  -5  5
 5(0   -1  0)
   25   0  5
```

// Если функция цели перестает зависеть от u_i, то её можно исключить из рассмотрения.

31. Метод Вульфа.

Этот метод основан непосредственно на рез-ах теоремы Кнута-Такера в дифференциальной форме. (1) $\min\left\{f(x)=\vec{c}^{\,t}\vec{x}^{\,t}D\vec{x}\,/\,A\vec{X}=\vec{b},\vec{x}\geq 0\right\}$

$$A=\left\{a_{ij}\right\}_{m\times n}\qquad \vec{C}=\begin{pmatrix}c_1\\c_2\\\dots\\c_n\end{pmatrix}\qquad \vec{X}=\begin{pmatrix}x_1\\x_2\\\dots\\x_n\end{pmatrix}\qquad \vec{b}=\begin{pmatrix}b_1\\b_2\\\dots\\b_n\end{pmatrix}$$

D – квадратная симметричная матрица размером N×N

Пусть D положительно определена или положительно полуопределена(для гарантии min),тогда совокупность необходимых и недостаточных условий.

$$(2).\begin{cases}A\vec{x}=\vec{b}\\2D\vec{x}-\vec{\upsilon}+A^T\vec{x}=-\vec{c}\\\vec{x}\geq 0\\\upsilon\geq 0\end{cases}$$

$(3).X_j\upsilon_j=0,\ j=\overline{1,n}$

Вместо (1) будем решать (2) и (3) используя дополнительное правило.

Если в текущем базисе есть переменная, X_j, то переменная V_j не должна включаться очередное ДБР и наоборот.

3 этапа решения задачи.

1 этап .- Определить, если это возможн, ДБР для $A\bar{x}=\bar{b}$

Решается (2 и (3) с целью определить совместимость $A\bar{x}=\bar{b}$

Для этого введем набор искусственных переменных.

$$\vec{\omega}=\begin{pmatrix}\omega_1\\\dots\\\omega_m\end{pmatrix};\vec{Z}^{(1)}=\begin{pmatrix}z_1^{(1)}\\\dots\\z_n^{(1)}\end{pmatrix};\vec{Z}^{(2)}=\begin{pmatrix}z_1^{(2)}\\\dots\\z_n^{(2)}\end{pmatrix};\mu$$

И вместо $A\bar{x}=\bar{b}$ будем решать.

$$(4.)\begin{cases}A\vec{x}+\vec{w}=\vec{b}\\2D\vec{x}=\vec{v}+A\vec{\lambda}+\vec{Z}^1-\vec{Z}^{(2)}+\mu\vec{c}=0\\\vec{x}\geq 0;\vec{v}\geq 0;\vec{Z}^1\geq 0;\vec{Z}^{(2)}\geq 0;\vec{w}\geq 0,\mu\geq 0\end{cases}$$

$(5.)\dots x_j v_j=0,\ j=\overline{1,n}$

В базис 1ДБР входят все w_i и часть переменных $Z^{(1)}{}_j, Z^{(2)}{}_j$

$\sum_{i=1}^{m} w_i \to \min$ при огранич. (4) и (5)

В числе НБП принудительно удерживаем все $\vec{v},\vec{\lambda},\mu$

В результате получим некоторое ДБР задачи (4), (5)

При этом,если все w_i уйдут из базы то (1) совместна.

2 этап - Из последнего достигнутого ДБР (4),(5) вычеркнем все столбцы с W_i и все столбцы с НБП $Z^{(1)}{}_j, Z^{(2)}{}_j$.Остановимся в базисе переменного $Z^{(1)}{}_j, Z^{(2)}{}_j$ перенумеруемся заново. На втором этапе решаем min задачу вида.

$$\sum_{i=1}^{n} Z_j \to \min$$

При этом принудительно держим в НБП переменную μ.Обязаны обеспечить выполнения ограничения (5)

3 этап.- Надо определить решение задачи для конкретно заданного в-ра $\vec{C}$

$\mu \to \min$

Обязательно выдерживаем ограничение (5).Если не получается сделать ни одной итерации с учетом (5),то функция исходной задачи не ограничена снизу

В противном случаи делаем альфа итерации.В результате получаем Мю1 и Мю 2.

$\begin{pmatrix} \bar{x}^1 \\ \vec{v}^1 \\ \vec{\lambda}^1 \\ \mu_1 \end{pmatrix}$ $\begin{pmatrix} \bar{x}^2 \\ \vec{v}^2 \\ \vec{\lambda}^2 \\ \mu 2 \end{pmatrix}$ по этими двумя точкам сможем определить решение при $\mu = 1$

Решение определяется по формуле.

$$\begin{pmatrix} \bar{x}^* \\ \vec{v}^* \\ \vec{\lambda}^* \\ \mu_1 \end{pmatrix} = \frac{\mu_2 - 1}{\mu_2 - \mu_1} \cdot \begin{pmatrix} \bar{x}^2 \\ \vec{v}^2 \\ \vec{\lambda}^2 \\ \mu 2 \end{pmatrix} + \frac{1 - \mu_1}{\mu_2 - \mu_1} \begin{pmatrix} \bar{x}^1 \\ \vec{v}^1 \\ \vec{\lambda}^1 \\ \mu_1 \end{pmatrix}$$

B)

$$\mu_2 < 1;$$

$$\begin{pmatrix} \bar{x}_x \\ \bar{v}_x \\ \bar{\lambda}_x \\ \mu = 1 \end{pmatrix} = \begin{pmatrix} \bar{x}^1 \\ \vec{v}^1 \\ \vec{\lambda}^1 \\ \mu_1 \end{pmatrix} + \frac{1 - \mu_1}{\mu_2 - \mu_1} \left(\begin{pmatrix} \bar{x}^2 \\ \vec{v}^2 \\ \vec{\lambda}^2 \\ \mu 2 \end{pmatrix} - \begin{pmatrix} \bar{x}^1 \\ \vec{v}^1 \\ \vec{\lambda}^1 \\ \mu_1 \end{pmatrix} \right)$$

Надо считать в общем случаи компоненты. $\bar{\lambda} = \begin{pmatrix} \lambda_1 \\ \dots \\ \lambda_m \end{pmatrix}$ м.б. любого знака,а

следовательно в остальных методах(где $\bar{\lambda} \geq 0$) Мю

заменить. $\lambda_i = \lambda_i^{(1)} - \lambda^{(2)}{}_i, \lambda_i^{(1)} \geq 0; \lambda^{(2)}{}_i \geq 0$.

32. Метод кусочно-линейной аппроксимации.

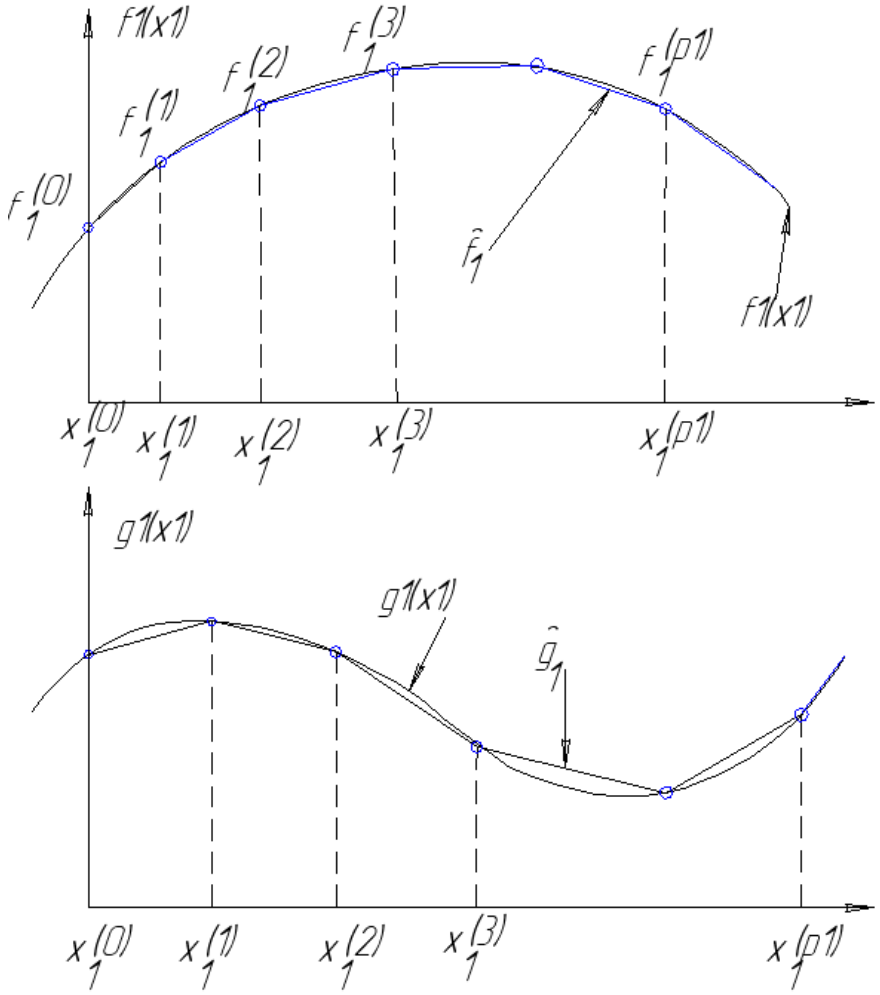
Этот метод надежен,если все функции являются сепарабельными, причем если они выпуклые,а ср. ц. вогнутая,то метод позволяет найти глобальный экстремум, иначе гарантировано нах-ие локального экстремума,в качестве приближения к истинному решению.

1. $f(x) = \sum_{j=1}^{n} f_j(x_j) \to \max(\textit{ксенараб} - \textit{функция})$
2. $\sum_{j=1}^{n} g_{ij}(x_i) = b_i, i = 1, m$
3. $x_j \geq 0,\ j = \overline{1,n}$

Рассмотрим общ. Схему метода:

Применительно к X1 проведем аппроксимацию для любой функции,которая зависит от X1 в виде линейной комбинации значений этих функций.

1. определим отрезок взаимных значений X1.
2. Разобьем этот отрезок на произвольное число подинтервалов.



На любом из подинтервалов справедливо опред-ии вып-ой и вогнутой функции с исп. ВЛК(Вып. Мин. комбинации)

$h(Ax_1^{(3)} + (1-\lambda)x_1^{(4)}) \leq \lambda h(x_1^{(3)}) + (1-\lambda) f(x_1^{(4)})$

Применительно к $f_1(x_1)$

$$\begin{cases} \hat{f}_1 = \sum_{k=0}^{p_1} f_1^{(k)} \lambda^k{}_1 \\ \hat{g}_{11} = \sum_{k=0}^{p_1} g_{11}{}^{(k)} \lambda^k{}_1 \qquad \lambda - \textit{неизвестныепеременные.} \\ \sum_{k=0}^{p_1} \lambda^k{}_1 = 1 \\ \lambda^{(k)} \geq 0, k = \overline{0, p_1} \end{cases}$$

Правило.

Для переменной x_1 в любом ДБР заменяющей линейной задаче отличными от Ф м.б. одна или две переменных $\lambda_1^{(л)}$,соседние по номеру к,а остальные д.б.=1

Мы должны принудительно обеспечивать правило для любых переменных x_j

С такой моделью мы получим задачу.

$$\begin{cases} f(\overline{\lambda}) = \sum_{j=1}^{n} \sum_{k=0}^{p_j} f_j{}^k \lambda_j{}^k \rightarrow \max \\ \sum_{j=1}^{n} \sum_{k=0}^{p_j} g_{ij}{}^k \lambda_j{}^k = b_i, i = \overline{1,m} \\ \sum_{k=0}^{p_1} \lambda_j{}^{(k)} = 1, \; j = \overline{1,n} \\ \lambda_j{}^k \geq 0, \; j = \overline{1,n}.....,k = \overline{o, p_i} \end{cases}$$

«+» эту зад. М. решать как обычную линейную задачу, проверил на любой итерации правила.

«-»Размерность зад. Много больше,поэтому,если X_j выходит в исх. зада. Всюду линейна($C_j X_j$ и $A_{ij}X_j$),то по этой переменной не обязательно проводить аппроксимацию,а прямо так включать ее в задачу.

Условия остановки.

1. $\overline{\lambda}^*$ -оптимальное решение с учетом признаков.
2. Раньше,чем получим $\overline{\lambda}^*$,если не удается выполнить требования правила,тогда $\lambda_{тек} \approx \overline{\lambda}^*$ принимается за приближенное решение тек. Решения уточняется,проводя разбиение интервалов значения, X_j на более мелкие интервалы в окрестности полученного решения.

Переход к исход.

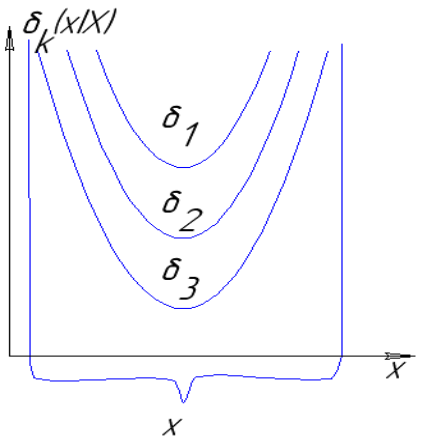
$$X_j = \sum_{k=0}^{p_j} x_j{}^{(k)} \lambda_j{}^{(k)} ;....\overline{j=1,n}$$

35. Методы Штрафных функций.

Min f(x) решение такой задачи заменяется на решение бесконечной последовательности задач безусловной оптимизации следующего вида: $\min\left(f(x)+\lim\limits_{k->\infty}\delta_k(\bar{x}\mid X)\right)$

$\lim\limits_{k\to\infty}\delta_k(\bar{x}\mid X)=\delta(\bar{x}\mid X)=\begin{cases}0,\forall\bar{x}\in X\\+\infty,\forall x\notin X\end{cases}$ - идеальная функция штрафа. В зависимости от вида исходной задачи существуют 2 метода штрафных функций.

1) Метод внутренних штрафных функций (барьерных функций).



$\delta(\bar{x}\mid X)=\begin{cases}0,\forall\bar{x}\in X\\+\infty,\forall x\notin X\end{cases}$

Находим минимум с гарантией, т.к f(x) и $\delta_k(\bar{x}\mid X)$ - выпуклые.

Задача: min f(x) где f(x) – выпуклая функция

$X:\tilde{g}_i(\bar{x})\ge 0\ i=1,m$ - вогнутые функции т.е $-\tilde{g}_i(\bar{x})\le 0$ - выпуклая.

$F_k(x,r_k)=f(\bar{x})-r_k\sum\limits_{i=1}^{m}\ln\tilde{g}_i(x)\to\min$ при росте

$k: r_k\underset{k\to\infty}{\to}0+,\ F_k(\bar{x},r_k)=f(\bar{x})+r_k\sum\limits_{i=1}^{m}\frac{1}{\tilde{g}_i(x)}\to\min$

Алгоритм:

1) $x^{(0)}\in X_{внутр}$ *т.е* $\tilde{g}_i(x^{(0)})>0\ i=1,m$, (??? Некоторое условие остановки???)

2) Решаем последовательность задач безусловной оптимизации

$F_k(\bar{x},r_k)=f(\bar{x})-r_k\sum\ln\tilde{g}_i(\bar{x})\to\min$ начинаем из точки $\tilde{\tilde{x}}^{(k-1)}$ для поиска приближения $\tilde{\tilde{x}}^{(k)}$

$\tilde{\tilde{x}}^{(k)},\tilde{\tilde{x}}^{(k-1)},...,\tilde{\tilde{x}}^{(0)}$

3) $\lim\limits_{k\to\infty}\tilde{\tilde{x}}^{(k-1)}=\tilde{\tilde{x}}*$

2) Метод внешних штрафных функций

Этим методом удобно решать следующие задачи

$\begin{cases}f(\bar{x})\to\min\ \ f(x)-выпуклая\\ \tilde{g}_i(x)\le 0\ i=1,m\\ \tilde{g}_i(\bar{x})=0\ i=m+1,l\end{cases}$ $\delta_k(\bar{x}\mid X)=\begin{cases}0,\ \forall x\in X\\>0,\ \forall\bar{x}\notin X\end{cases}$ $\delta_k(\bar{x}\mid X)=r_k\Phi_k(\bar{x}\mid X)\ r_k\underset{k\to\infty}{\to}\infty$

Решение задачи начинается из недопустимой точки $\bar{x}\notin X$.

Сушествует 2 вида функции штрафа:

$\Phi_1(\bar{x}\mid X)=\sum\limits_{i=1}^{m}\left(\tilde{g}_i^{+}(x)\right)^2+\sum\limits_{i=m+1}^{l}\left(\tilde{g}_i(x)\right)^2$

$\Phi_2(\bar{x}\mid X)=\sum\limits_{i=1}^{m}\tilde{g}_i^{+}(x)+\sum\limits_{i=m+1}^{l}|\tilde{g}_i(x)|$

$\tilde{g}_i^{+}(x)$-срезка- функция для $\tilde{g}_i(x)$ отсекающая все отрицательные значения: $\tilde{g}_i^{+}(x)=\begin{cases}0,\tilde{g}_i(x)\le 0\\ \tilde{g}_i(x)\end{cases}\mid\tilde{g}_i^{+}(x)=\max\{0,\tilde{g}_i(x)\}$

В методе внутренних штрафных функций мА начинаем из $x\in X$ и никогда из нее не выходим => можем прекратить работу алгоритма на любой итерации, но мы должны для любого ограничения вычислять частные производные и обеспечивать $x\in X_{внутр}$.

В методе внешних штрафных функций работа начинается из недоступной точки и решаем задачу $F_k(\bar{x},r_k)=f(\bar{x})+r_k\Phi_k(\bar{x}\mid X)\to\min;\ r_k\underset{k\to+\infty}{\to}+\infty$, поэтому решение получается

в пределе при больших к. но зато при любой итерации нужно обрабатывть только нарушаемые ограничения $\tilde{g}_i\left(\bar{x}^{(k)}\right) > 0$

Алгоритм:

1) $\bar{x}^{(0)} \notin X$, задаем условие остановки

2) решаем последовательность задач бузусловной оптимизации

$$F_k\left(\bar{x}, r_k\right) = f(x) + r_k\left(\sum_{i=1}^{m}\left(\tilde{g}_i^{+}(x)\right)^2 + \sum_{i=m+1}^{l}\left(\tilde{g}_i(x)\right)\right) \to \min \; r_k \underset{k\to+\infty}{\to} +\infty$$

$$\lim_{k\to\infty} \tilde{\bar{x}}^{(k)} = \bar{x}^*$$

36. методы случайного поиска

В этих задачах намеренно вводится элемент случайного выбора исходных параметров. $f(x) \to \min \ \bar{x} \in X$ $\bar{x}^{(k+1)} = x^{(k)} + d_k \bar{\zeta}^{(k)}, k = 0,1,2...$ $\bar{\zeta}^{(k)}$ - вектор случайных величин.

1) Медод случайного поиска с возвратом при неудачном шаге.

$\bar{\zeta}^{(k)}$, вычисляем $\bar{y}^{(k+1)} = \bar{x}^{(k)} + d\bar{\zeta}^{(k)}$ если $f\left(\bar{y}^{(k+1)}\right) < f\left(\bar{x}^{(k)}\right) u \ y^{(k+1)} \in x$ то шаг считаем удачным и $\bar{x}^{(k+1)} = \bar{y}^{(k+1)}$ иначе шаг считаем неудачным и остаемся в $\bar{x}^{(k)}$

Условие остановки: если $\bar{x}^{(k)} = \bar{x}^{(k+1)} = \bar{x}^{(k+2)} = .. = \bar{x}^{(k+N)}$ то останавливаем алгоритм и считаем $\bar{x}* = \bar{x}^{(k)}$

2)алгоритм наилучшей пробы

Многократная случайная выборка $\bar{\zeta}^{(1)}, \bar{\zeta}^{(2)},..., \bar{\zeta}^{(m)}$ - среди них выбирается наилучшая по правилу: $f\left(\bar{x}^{(k)}\right) + d\bar{\zeta}^{(i)} = \min\limits_{1 \le i \le m} f\left(\bar{x}^{(k)} + d\bar{\zeta}^{(i)}\right)$ и вычисляем $\bar{x}^{(k+1)} = \bar{x}(k) + d\bar{\zeta}^{(i)}, \ d > 0, m > 1$

3)алгоритм статического градиента.

m, $\bar{\zeta}^{(1)}$, $\bar{\zeta}^{(2)}$,..., $\bar{\zeta}^{(m)}$, оцениваем: $\Delta f_i^{(k)} = f\left(\bar{x}^{(k)} + \gamma\bar{\zeta}^{(i)}\right) - f\left(\bar{x}^{(k)}\right), \ i = 1, m$.

Усредняем $\bar{\zeta}^{(i)}$ по формуле $p^{(k)} = \frac{1}{\gamma}\sum_{i=1}^{m} \Delta f^{(k)} \bullet \bar{\zeta}^{(i)}$ (р- статический градиент дельта ф это веса случайных векторов). Делаем шаг в направлении статического градиента $\bar{x}^{(k+1)} = \bar{x}^{(k)} + ap^{(k)}$

4)алгоритм случайного поиска с покоординатным обучением.

В предыдущих алгоритмах не учитывались результаты предидущих ходов, разумно былобы учитывать , т.е целенаправленно изменить распространение случайных волн => такой алгоритм является само обучающимся.

$$\bar{\xi} = \bar{\xi}(\bar{w}) \ \bar{\xi}(\bar{w}) = \left(\bar{\xi}_1(\bar{w}_1),..., \bar{\xi}_n(\bar{w}_n)\right)^T; \ \xi_i = \begin{cases} +1 c \ вероятностью \ p_j \\ -1 c \ вероятностью \ (1 - p_{j)} \end{cases} \quad j = 1, n$$

$$p_j = \begin{cases} 0, \ w_j < -1 \\ 0.5 \bullet (1 + w_j); w_j \in [-1,1] \\ 1, \ w_j > 1 \end{cases}$$

Алгоритм:

1) На первом шаге нет обучения т.е $\bar{x}^{(0)}, d, \bar{x}^{(1)} = \bar{x}^{(0)} + d\bar{\xi}\left(\bar{w}^{(0)}\right), \bar{w}^{(0)} = (0,...,0)^T$
2) На 2-м шаге тоже нет обучения. $\bar{x}^{(2)} = \bar{x}^{(1)} + d\bar{\xi}\left(\bar{w}^{(1)}\right), \bar{w}^{(1)} = (0,...,0)^T$
3) Начиная с 3-го шага выполняется самообучение алгоритма
   $\bar{x}^{(k+!)} = \bar{x}^{(k)} + d\xi\left(\bar{w}^{(k)}\right), k > 1$, известны: $x^{(0)}, x^{(1)},..., x^{(k+1)}, \bar{w}^{(k)}$, найдем
   $\bar{w}_j^{(k+1)} = \beta w_j^{(k)} - \delta sign\left[\left(f\left(x^{(k+1)}\right) - f\left(x^{(k)}\right)\right) \bullet \left(x_j^{(k)} - x_j^{(k+1)}\right)\right], \ j = 1, n$ вычислим p_j и переходим в точку $\bar{x}^{(k+2)} = \bar{x}^{(k+1)} + d\xi\left(\bar{w}^{(k+!)}\right)$

Выбираем сами параметры:

$\beta \ge 0$ - параметр интенсивности запоминания предидущего опыта(чем он выше тем сильнее память алгоритма)

$\delta \ge 0$ - параметр скорости обучения (чем он выше тем быстрее обучаемость)

Если хоть один из параметров = 0 то алгоритм не обучается

4) Если переход $\bar{x}^{(k-1)} \to \bar{x}^{(k)}$ удачен то на очередном шаге вероятность выбора направления $\left(\bar{x}^{(k)} - \bar{x}^{(k-1)}\right)$ увеличивается иначе уменьшается .

Билет 37. Постановка общей задачи линейного программирования. Примеры задач.

Общий вид задачи линейного программирования(ЗЛП):

$$\bar{x} = (x_1, x_2, \dots, x_n)^T$$

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

$$(*)\begin{cases} \sum_{j=1}^{n} a_{ij} x_j \le b_i \;\; , i = \overline{1,k} \\ \sum_{j=1}^{n} a_{ij} x_j = b_i \;\; , i = \overline{k+1,m} \\ x_j \ge 0 \;, \quad j = \overline{1,t} \;\; (t \le n) \end{cases}$$

Ту же задачу можно записать в матричном виде:

$$f(\bar{x}) = c^T \bar{x} \to \min(max), A\bar{x} \le b, \tilde{A}\bar{x} = \tilde{b}, x_j \ge 0 \; (j = \overline{1,t})$$

Где A – матрица k на n, $\tilde{A}$ - матрица (m-k) на n, c – вектор из n элементов, b – вектор из k элементов, $\tilde{b}$ – вектор из (m-k) элементов.

Канонический вид ЗЛП:

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

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j = b_i \;, & i = \overline{1,m} \\ x_j \ge 0, & j = \overline{1,n} \end{cases}$$

Пример: задача о переменном рационе.

x_j — количество j — го вида продукта в рационе

a_{ij} — количсетво i — го вещества в j — ом продукте

b_i — суточная потребность в i — ом веществе

c_j — стоимость j — го вида продукта

Постановка задачи: составить такой рацион потребления различных продуктов, в котором содержится как минимум суточная потребность в каждом из веществ, минимизируя стоимость.

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

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j \ge b_i \;, & i = \overline{1,m} \\ x_j \ge 0, & j = \overline{1,n} \end{cases}$$

Билет 38. Свойства решений ЗЛП.

Имеем задачу ЛП в общем виде.

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

$$(*)\begin{cases} g_i(\bar{x}) = \sum_{j=1}^{n} a_{ij} x_j \le b_i \;\; , i = \overline{1,k} \\ g_i(\bar{x}) = \sum_{j=1}^{n} a_{ij} x_j = b_i \;\; , i = \overline{k+1,m} \\ x_j \ge 0 \;, \quad j = \overline{1,t} \;\; (t \le n) \end{cases}$$

Теорема 1 об области определения ЗЛП.

X – область, определяемая ограничениями (*).
X - выпуклое замкнутое множество.
Все функции g_i – являются линейными, значит и выпуклыми. По теореме 4 о выпуклых функциях область X данной задачи является выпуклой.
Теорема 4 о выпуклых функциях: Если функции g_i – выпуклые, то область
$X = \{\bar{x} | g_i(\bar{x}) \le b_i \ , \bar{x} \ge 0\}$ является выпуклым множеством.
Покажем замкнутость:
Сформируем последовательность
$\bar{x}^{(1)}, \bar{x}^{(2)}, \dots - \text{сходящуюся к } \bar{x} - \text{какой либо точке } X, \text{т.е.} \lim_{r\to\infty} \bar{x}^{(r)} = \bar{x}$, причем все эти точки принадлежат множеству X.
Согласно свойству непрерывности скалярного произведения: $\lim_{r\to\infty}(A^{(i)}, \bar{x}^{(r)}) = (A^{(i)}, \bar{x})$ (выр 1),
Где $A^{(i)} = (a_{i1}, a_{i2}, \dots, a_{in}), \text{для } i = \overline{1,m}$
С учетом выр 1, получаем предельную зависимость:
$(A^{(i)}, \bar{x}) = b_i \ , i = \overline{1,k}$
$(A^{(i)}, \bar{x}) = b_i \ , i = \overline{k+1,m}$
Но в качестве точки $\bar{x}$, может быть выбранна любая точка X, а значит и граничная точка, т.к. никаких ограничений на последовательность $\bar{x}^{(r)}$ мы не накладываем. Но множество, содержащие все свои граничные точки замкнуто, а значит и множество X - замкнуто.
Замечания:
Выражения $(A^{(i)}, \bar{x}) \le b_i$, определяют полупространство, а $(A^{(i)}, \bar{x}) = b_i$ – гиперплоскость, постольку множество планов ЗЛП, определяемое системой (*), является пересечением конечного числа полупространств и гиперплоскостей. Такая область называется выпуклым многогранным множеством, если оно ограниченно – выпуклым многогранником.
Понятие выпуклого многогранника имеет связь с понятием выпуклой оболочки, т.е. выпуклый многогранник представляет собой выпуклую оболочку своих крайних точек, т.е. множество точек
$\bar{x} = \sum_{i=1}^{N} \mu_i \bar{x}^{(i)}, \quad \text{где} \sum_{i=1}^{N} \mu_i = 1, \quad 0 \le \mu_i \le 1, \quad \text{а} \quad \bar{x}^{(i)} - \text{крайние точки многогранника}$
Крайняя точка выпуклого многогранного множества называется вершиной(опорным планом). Такая точка будет вершиной в случае, если среди ограничений (*) – ровно n линейнонезависимых ограничений будут выполнены как строгие равенства.
Вершина называется вырожденной, если в ней выполняется больше чем n ограничений как равенств. Число вершин многогранника конечно, значит и число опорных планов конечно.
Опорный план представляет собой выпуклую линейную комбинацию(ВЛК) вершин многогранника, иными словами опорные планы представляют собой основу(опорные точки), дающие возможность построить любой допустимый план задачи, потому эти точки и названы опорными.
По теореме о крайней точке: если множество выпуклое, замкнутое и ограниченное, то существует хотя бы одна его крайняя точка(опорный план).
Теорема 2 об оптимальности:
Линейная форма $f(\bar{x}) = c^T\bar{x}, \text{определенная на выпуклом многограннике } A\bar{x} \le b, \bar{x} \ge 0$ достигает своего экстремума в вершине этого многогранника.
Док-во: Пусть $x^{(1)}, x^{(2)}, \dots, x^{(N)}$ – все вершины многогранника X, на котором линейная форма определена и
$\bar{x}^* - \text{точка многогранника, в которой линейная форма достигает своего минимума}$. Также эта точка – это ВЛК вершин многогранника:
$\bar{x}^* = \sum_{i=1}^{N} \mu_i \bar{x}^{(i)}, \ \mu_i \ge 0, \sum_{i=1}^{N} \mu_i = 1$
Тогда
$f(\bar{x}^*) = \sum_{i=1}^{N} \mu_i f(\bar{x}^{(i)}), \text{выберем наименьшее } f(\bar{x}^{(i)}), \text{например в точке } \bar{x}^{(i)}, \text{тогда}$

$f(\overline{x}^*) \geq \sum_{i=1}^{N} \mu_i f(\overline{x}^{(i)}) = f\left(\overline{x}^{(i)}\right)$, **но поскольку** $f(\overline{x}^*)$ - наименьшее значение линейной формы, то оно должно совпадать с $f(\overline{x}^1)$, значит в какой-то вершине многогранника минимум достигается с гарантией.

Теорема 3:

Для того, чтобы ЗЛП была разрешимой необходимо и достаточно выполнение двух условий:

1) множество планов задачи не пусто;

2) линейная форма ограничена на этом множестве сверху (снизу), для задачи поиска максимума (минимума).

Выводы:

1) непустое мн-во планов задачи имеет конечное число опорных планов;

2) любая разрешимая задача имеет оптимальное решение среди опорных планов и может быть найдено за конечное число шагов;

3) множество точек, в которых линейная форма имеет постоянное значение, есть гиперплоскость, и гиперплоскости, соответствующие различным значениям линейной формы параллельны между собой;

4) локальный экстремум является одновременно глобальным экстремумом линейной формы на мн-ве планов задачи.

Билет 39. Двойственные ЗЛП и их свойства.

Каждой ЗЛП можно поставить в соответствие другую ЗЛП, которую называют двойственной по отношению к 1-й задаче.

Пусть прямая задача задана в общем виде:

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

$$(*)\begin{cases} g_i(\vec{x}) = \sum_{j=1}^{n} a_{ij} x_j \le b_i \ \ , i = \overline{1,k} \\ g_i(\vec{x}) = \sum_{j=1}^{n} a_{ij} x_j = b_i \ \ , i = \overline{k+1,m} \\ x_j \ge 0 \ , \quad j = \overline{1,t} \ \ (t \le n) \end{cases}$$

Двойственная задача (или сопряженная) будет записана следующим образом:

$$\tilde{f}(\vec{x}) = \sum_{i=1}^{m} b_i y_i \to min$$

$$(*)\begin{cases} \sum_{i=1}^{m} a_{ij} y_i \ge c_j \ \ , j = \overline{1,t} \ , \mathrm{t} \le \mathrm{n} \\ \sum_{i=1}^{m} a_{ij} y_j = c_j \ \ , j = \overline{t+1,n} \\ y_i \ge 0 \ , \quad i = \overline{1,k} \ \ (k \le m) \end{cases}$$

Правила формирования двойственной задачи:

1) вид экстремума меняется на противоположный.

2) векторы b, c - меняются местами.

3) матрица условий A* двойственной задачи – образуется транспонированием матрицы условий исходной задачи.

4) j – е условие двойственной задачи будет неравенством, если на j – ю переменную прямой задачи наложено требование неотрицательности, в противном случае ограничение будет равенством.

5) на i – ю переменную двойственной задачи должно быть наложено условие неотрицательности, если i - е, ограничение прямой задачи является неравенством.

Для канонической формы задачи: (зад 1)

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

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j = b_i \ , \quad i = \overline{1,m} \\ x_j \ge 0, \quad j = \overline{1,n} \end{cases}$$

Двойственная будет выглядеть: (зад 2)

$$\tilde{f}(\vec{x}) = \sum_{i=1}^{m} b_i y_i \to min$$

$$\sum_{i=1}^{m} a_{ij} y_i \ge c_j \ , \quad j = \overline{1,n}$$

Связь между прямой и двойственной к ней задачами (задачами двойственной пары):

Лемма 1:

Если $\bar{x} = (x_1, x_2, \dots, x_n)$ **и** $\bar{y} = (y_1, y_2, \dots, y_m)$ являются произвольными допустимыми планами задач 1 и 2, то функции цели этих задач связаны $f(\bar{x}) \le \tilde{f}(\bar{y})$.

$$\sum_{i=1}^{m} a_{ij}y_i \geq c_j, \quad j=\overline{1,n}, \text{значит} \sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} x_j \sum_{i=1}^{m} a_{ij}y_i = \sum_{i=1}^{m} y_i \sum_{j=1}^{n} a_{ij}x_j$$

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \text{значит} \sum_{j=1}^{n} c_j x_j \leq \sum_{i=1}^{m} b_i y_i \Rightarrow f(\overline{x}) \leq \tilde{f}(\overline{y}).$$

Лемма 2:

Если для некоторых планов

$\overline{x}^*$ и $\overline{y}^*$: $f(\overline{x}^*) = \tilde{f}(\overline{y}^*)$, **то эти планы являются оптимальным планами.**

В силу леммы 1 можно записать $f(\overline{x}) \leq f(\overline{y}^*)$, но по утверждению нашей леммы получаем: $f(\overline{x}) \leq f(\overline{x}^*)$, что говорит о том, что выбранная точка является точкой масимума. Оптимальность плана $\overline{y}^*$ доказывается аналогично.

Лемма 3:

Если линейная форма $\tilde{f}(\overline{y})$ –не ограниченна снизу на множестве своих планов, то прямая задача не имеет ни одного плана.

Из условия леммы следует, что существует такая последовательность планов $\{\overline{y}^{(r)}\}$ двойственной задачи, что $\lim_{r\to\infty} \tilde{f}\left(\overline{y}^{(r)}\right) = -\infty$

Предположим, что прямая задача имеет некоторый план $\overline{x}$, тогда по лемме 1: $f(\overline{x}) = -\infty$, что невозможно при любом плане с конечными составляющими, значит множество планов прямой задачи пусто.

Теореме 1: (первая теорема двойственности)

Если одна из задач двойственной пары имеет оптимальное решение, то и другая задача разрешима. При этом для оптимальных планов имеет место равенство $f(\overline{x}^*) = \tilde{f}(\overline{y}^*)$

Следствия:

1) для разрешимой одной из задач двойственной пары необходимо и достаточно, чтбы каждая из задач имела хоть один план.

2) для того, чтобы одна из задач двойственной пары имела планы, а множество планов другой было пусто, необходимо и достаточна неограниченность функции цели первой задачи на множестве ее планов.

3) для оптимальности планов двойственной пары необходимо и достаточно выполнение равенства $f(\overline{x}^*) = \tilde{f}(\overline{y}^*)$

Теорема2: (вторая теорема двойственности)

Если оптимальный план одной из задач обращает i – е ограничение этой задачи в строгое неравенство, то l – я компонента оптимального плана другой задачи равна нулю.

Если же j – я компонента оптимального плана одной задачи положительна, то оптимальный план двойственной задачи обращает j – е ограничение в строгое равенство.

Т.е. выполняются условия дополняющей нежесткости:

$$\left(b_i - \sum_{j=1}^{n} a_{ij}x_j^*\right)y_i^* = 0, \quad i=\overline{1,k}$$

$$\left(\sum_{i=1}^{m} a_{ij}y_i^* - c_j\right)x_j^* = 0, \quad j=\overline{1,t}$$

Билет 40. Идея метода последовательного улучшения плана, признак оптимальности.

Решение ЗЛП – набор переменных x_j, который удовлетворяет всем ограничениям задачи за исключением ограничений, накладываемых на знак самих переменных.

Допустимое решение (ДР) – решение с неотрицательными компонентами.

Базис (Б) – набор из m переменных(по числу ограничений ЗЛП), которые выражены через остальные переменные. Эти m переменных называются базисными (БП), а остальные – небазисными (НБП).

Базисное решение (БР) – такое решение, в котором всем небазисным переменным присваиваются нулевые значения, а значения базисных переменных вычисляются.

Допустимым базисным решением (ДБР) называется такое базисное решение, которое одновременно является и допустимым.

Идея метода последовательного улучшения плана:

Сначала находим ДБР, которое совпадает с вершинной выпуклого многогранного множества, определяем является ли это решение оптимальным. Если нет, то выводим из базиса один вектор и заменяем его другим вектором, получая таким образом новый базис. Его также исследуем на оптимальность. Поскольку количество вершин многогранного множества конечно, то и подобный процесс конечен, и в результате его либо находится экстремум, либо доказывается несовместность ограничений, либо доказывается неограниченность ф-ии цели снизу (для минимизации) или сверху (для максимизации).

Пусть ЗЛП задана в каноническом виде:

$f(\overline{x}) = c^T\overline{x} \to min$ (3)

$\sum_{j=1}^{n} A_j x_j = b$, (1)

$x_j \geq 0$, $j = \overline{1,n}$ (2)

Где $A_j = (a_{1j}, a_{2j}, \dots, a_{mj})^T -$ **вектор состоящий из элементов $j-$ го столбца матрицы A**

В общем случае будем предполагать, что ранг матрицы А равен m.

Предположим, что нам известно ДБР задачи , т.е. задан набор базисных переменных $\overline{x}_k^{(r)} = \left(x_1^{(r)}, x_2^{(r)}, \dots, x_m^{(r)}\right)$, где индекс r просто указывает на принадлежность данного параметра базису. Соответственно задан и базисный набор векторов $A_j^{(r)}$**, составляющих базис векторов** $A^{(r)} = \left\{A_1^{(r)}, A_2^{(r)}, \dots, A_m^{(r)}\right\}$, в этом случае произвольный вектор условий A_j может быть разложен по базису

$A^{(r)}: A_j = \sum_{k=1}^{m} A_k^{(r)} q_{kj}$, $j = \overline{1,n}$, **где $q_{kj}-$ это составляющие разложения векторов A_j** по векторам базиса при вектор $A_k^{(r)}$, находящийся в k – й позции базиса.

Подобной же операции можно подвергнуть вектор b: $b = \sum_{k=1}^{m} A_k^{(r)} q_{k0} = A_0$,где индекс 0 показывает, что эти составляющие расположены в столбце, соответствующем вектору ограничений.

Т.е. можно написать общую формулу: $A_j = \sum_{k=1}^{m} A_k^{(r)} q_{kj}$, $j = \overline{0,n}$ (4)

Кроме того, при подобном разложении всех векторов функция цели приобретает вид :

$f(\overline{x}) = -\sum_{j=1}^{n} \Delta_j x_j$, где $\Delta_j = z_j - c_j$, $j = \overline{1,n}$ (5)

$$z_j = \sum_{k=1}^{m} c_k^{(r)} q_{kj}, \qquad (6)\ j = \overline{1,n}$$, **где индекс r при c указывает, что это коэффициент при БП**

Все эти параметры определяются однозначно, при выборе того или иного базиса. Используя параметры Δ_j можно сформулировать признак оптимальности ДБР в виде теоремы.

Теорема: (признак оптимальности)

ДБР является оптимальным, если $\Delta_j \leq 0$, $j = \overline{1,n}$

Обозначим символом $\overline{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$ – опорный план, для которого выполняется

$\Delta_j = z_j - c_j \leq 0$, $j = \overline{1,n}$ **, т.е.** $z_j \leq c_j$ (7)

А символом $\overline{x}$ - обозначим произвольный опорный план, не совпадающий с планом $\overline{x}^*$. Будем исследовать произвольный опорный план. Для него должны выполняться условия (1), (2); причем условие (1) запишем в несколько другой форме, воспользовавшись выражением (4). В этом случае

получим: $\sum_{j=1}^{n} x_j \left(\sum_{k=1}^{m} q_{kj} A_k^{(r)}\right) = b, \text{или } \sum_{k=1}^{m} \left(\sum_{j=1}^{n} q_{kj} x_j\right) A_k^{(r)} = b$ (8). Но план $\overline{x}^*$ также удовлетворяет условиям (1), поэтому можно записать $b = \sum_{k=1}^{m} x_k^* A_k^{(r)}$ (9) – поскольку это ДБР, т.е. все НБП равны 0.

Сравнивая выражения (8) и (9) делаем вывод, что $\sum_{j=1}^{n} x_j q_{kj} = x_k^*$ (10)

Используя выражения (6), (7), (10), можно записать применительно к плану $\overline{x}^*$: $c_j \geq z_j, \text{при } j = \overline{1,n}$

Дополним обе части неравенства на x_j и сложим по всем $j = \overline{1,n}$, тогда слева получим $f(\overline{x})$ для произвольного плана ЗЛП, а справа произойдет перегруппировка слагаемых.

$$f(\overline{x}) = \sum_{j=1}^{n} c_j x_j \geq \sum_{j=1}^{n} z_j x_j = \sum_{j=1}^{n} \left(\sum_{k=1}^{m} c_k^{(r)} q_{kj}\right) x_j = \sum_{k=1}^{m} \left(\sum_{j=1}^{n} x_j q_{kj}\right) c_k^{(r)} = \sum_{k=1}^{m} c_k^{(r)} x_k^* = f(\overline{x}^*)$$

Следовательно план $\overline{x}^*$ является оптимальным, т.к. $f(\overline{x}) \geq f(\overline{x}^*)$.

Билет 41. Алгебраическое обоснование метода последовательного улучшения плана.

Рассмотрим ЗЛП в канонической форме

$f(\overline{x}) = c_1x_1 + c_2x_2 + \cdots + c_nx_n \to max$ (1)

$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$

$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$

…… (2)

$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$

$x_j \geq 0 \ , \ j = \overline{1,n}$ (3)

Пусть X – область заданная ограничениями (2), (3) не пуста и ранг системы условий (2)равен m. Будем считать m<n. Если m=n, то система условий (2) является крамеровской и, следовательно, область X определения задачи состоит из единственной точки, которая и является решением задачи. Пусть нам известен начальный опорный план задачи (1), (2), (3)

$\overline{x}^{(0)} = (x_1, x_2, \ldots, x_n)$, положим, что первые $n - m$ компонент этого плана небазисные (НБП), а последние m компонент – базисные (БП). Обозначим n-m=k. Разрешим систему (2) относительно базисных переменных:

$x_{k+1} = \beta_1 - (\alpha_{11}x_1 + \cdots + \alpha_{1k}x_k)$

$x_{k+2} = \beta_2 - (\alpha_{21}x_1 + \cdots + \alpha_{2k}x_k)$

…… (4)

$x_{k+m} = \beta_m - (\alpha_{m1}x_1 + \cdots + \alpha_{mk}x_k)$

В начальном опорном плане $\overline{x}^{(0)}$ НБП равны нулю, при этом базисные переменные принимают значения $x_{k+i} = \beta_i \geq 0 \ , \ i = \overline{1,m}$, так как в противном случае имеем противоречие условиям (3), Таким образом начальный опорный план $\overline{x}^{(0)} = (0, \ldots, 0, \beta_1, \ldots, \beta_m)$. Заметим, что опорный план совпадает с некоторым ДБР системы (2). Подставим выражения БП через НБП в линейную форму (1): $f(\overline{x}) = \gamma_0 - (\gamma_1x_1 + \cdots + \gamma_kx_k)$ (5), на опорном плане $\overline{x}^{(0)}$ значение линейной формы γ_0

Перейдем к другому - «лучшему» опорному плану, базис которого отличается от предыдущего плана лишь одним вектором. Этот опорный план будет соседним по отношению к предыдущему опорному плану. Для этого выберем НБП x_j опрного плана $\overline{x}^{(0)}$ такую, что $\gamma_j < 0$, тогда выражения (4), (5) запишутся:

$x_{k+1} = \beta_1 - \alpha_{1j}x_j$

……

$x_{k+l} = \beta_l - \alpha_{lj}x_j$

…… (6)

$x_{k+m} = \beta_m - \alpha_{mj}x_j$

$f(\overline{x}) = \gamma_0 - \gamma_jx_j$

Введем в базис опорного плана $\overline{x}^{(0)}$ вектор, соответствующей переменной x_j, и выведем из базиса вектор, который будет указан ниже. Тогда будет построен новый опорный план $\overline{x}^{(1)}$, базис которого отличается от базиса предыдущего одним вектором. Это означает, что переменная x_j станет базисной, а какая то из БП станет НБП.

При увеличении значения переменной x_j от нуля значение функции цели будет увеличиваться, т.к. $\gamma_j < 0$. При этом изменятся и значения БП.

-если коэффициент $\alpha_{lj} < 0$, то при увеличении x_j увеличивается и значение БП x_{k+l}

-если коэффициент $\alpha_{lj} = 0$, то при увеличении x_j значение БП x_{k+l} не меняется

-если коэффициент $\alpha_{lj} > 0$, то при увеличении x_j уменьшается значение БП x_{k+l}

Рассмотрим БП опорного плана $\overline{x}^{(0)}$, для которых коэффициенты $\alpha_{lj} > 0$. Пусть при увеличении x_j первой обратилась в нуль переменная x_{k+i}, т.е. $x_{k+i} = \beta_i - \alpha_{ij}x_j = 0$ (7), а остальные БП при этом останутся положительными $x_{k+l} = \beta_l - \alpha_{lj}x_j > 0$ (8)

из (7) и (8) следует, что $\frac{\beta_i}{\alpha_{ij}} = \min_l\{\frac{\beta_l}{\alpha_{lj}}, \alpha_{lj} > 0\}$

Итак мы определили БП которая станет свободной: x_{k+i} и, соответственно, базисный вектор, который выводится из базиса опорного плана.

Найдем выражение БП нового опорного плана через НБП:

Из i – го уравнения системы (4):

$$x_{k+i} = \beta_i - (\alpha_{i1}x_1 + \cdots + \alpha_{ik}x_k), \text{получим } x_j = \frac{\beta_i}{\alpha_{ij}} - \left(\frac{\alpha_{i1}}{\alpha_{ij}}x_1 + \cdots + \frac{1}{\alpha_{ij}}x_{k+l} + \cdots + \frac{\alpha_{ik}}{\alpha_{ij}}x_k\right)$$

(9)

Подставим выражение (9) в остальные уравнения (4), получим:

$$x_{k+l} = \left(\beta_l - \frac{\alpha_{lj}\beta_i}{\alpha_{ij}}\right) - \left((\alpha_{i1} - \frac{\alpha_{i1}\alpha_{lj}}{\alpha_{ij}})x_1 + \cdots + (-\frac{\alpha_{lj}}{\alpha_{ij}})x_{k+l} + \cdots + (\alpha_{lk} - \frac{\alpha_{ik}\alpha_{lj}}{\alpha_{ij}})x_k\right)$$

(10)

Преобразуем таким же образом функцию цели:

$$f(\overline{x}) = \left(\gamma_0 - \frac{\gamma_j\beta_i}{\alpha_{ij}}\right) - \left(\left(\gamma_1 - \frac{\gamma_j\alpha_{i1}}{\alpha_{ij}}\right)x_1 + \cdots + \left(-\frac{\gamma_j}{\alpha_{ij}}\right)x_{k+l} + \cdots + \left(\gamma_k - \frac{\gamma_j\alpha_{ik}}{\alpha_{ij}}\right)\right)$$

(11)

Такой переход является одним шагом модифицированного Жорданова исключения.

Очевидно, что новый план является опорным, т.е все БП принимают неотрицательные значения, это прямо вытекает из способа выбора переменной, выводимой из базиса.

Очевидно, что функция цели на новом опорном плане, принимает значение:

$$f\left(\overline{x}^{(1)}\right) = \left(\gamma_0 - \frac{\gamma_j\beta_i}{\alpha_{ij}}\right)$$

Принимая во внимание, что $\gamma_j < 0,\ \beta_i > 0,\ a_{ij} > 0$, значение функции цели на новом плане увеличилось. Дальнейшее решение задачи аналогично переходу от (4), (5) к (9), (10), (11).

Замечания: Признаком оптимальности текущего плана, является положительность всех коэффициентов при БП в выражении функции цели через БП. Если в функции цели есть отрицательные коэффициенты при БП, а в выражении соответствующей такому коэффициенту БП, через НБП все коэффициенты неположительны, то это говорит о неограниченности линейной формы сверху.

Билет 42. Метод искусственного базиса.

Имеем ЗЛП в каноническом виде

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

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j = b_i, & i = \overline{1,m} \\ x_j \geq 0, & j = \overline{1,n} \end{cases}$$

Составим вспомогательную задачу:

$$f(\vec{\xi}) = \sum_{i=1}^{m} \xi_i \to min$$

$$\begin{cases} \xi_i = \sum_{j=1}^{n} a_{ij} x_j - b_i ; \xi_i \geq 0, & i = \overline{1,m} \\ x_j \geq 0, & j = \overline{1,n} \end{cases}$$

Очевидно, что для этой задачи вспомогательные переменные являются БП 1-го ДБР вспомогательной задачи. Линейная форма вспомогательной задачи ограничена снизу, поэтому задача разрешима. Решим эту задачу:

1) Оптимальное значение равно 0. Тогда последний опорный план вспомогательной задачи будет 1-м ДБР исходной задачи и исходная задача разрешима.

2) Оптимальное значение меньше 0. В этом случае исходная задача несовместна. Действительно, предположим, что удалось найти опорный план исходной задачи, иными словами нашелся такой план вспомогательной задачи, при котором все вспомогательные переменные равны 0. Но в этом случае значение функции цели вспомогательной задачи равно 0, что противоречит оптимальному решению.

Билет 43. М метод. Двойственный метод последовательного улучшения плана.

М метод является разновидностью симплекс метода. Имеем ЗЛП в канонической форме:

$$f(\vec{x}) = c_0 + \sum_{j=1}^{n} c_j x_j \to min$$

$$\begin{cases} \sum_{j=1}^{n} a_{ij} x_j = b_i, & i = \overline{1,m} \\ x_j \geq 0, & j = \overline{1,n} \end{cases}$$

Вводим искусственные переменные $\xi_i \geq 0$, берем их в качестве базисных и решаем ЗЛП вида

$$\begin{cases} \xi_i = b_i - \sum_{j=1}^{n} a_{ij} x_j, \ i = \overline{1,m} \\ \xi_i \geq 0, \ i = \overline{1,m}; x_j \geq 0, \ j = \overline{1,n} \\ f(\bar{x}, \bar{\xi}) = f(\bar{x}) + M \sum_{i=1}^{m} \xi_i \to min \end{cases}$$

М – некоторое очень большое положительное число. Главное, чтобы М было больше любого другого числа, встречающегося в задаче. В выкладках численное значение М не требуется. Когда хотя бы одно $\xi_i > 0$, то значение функции цели М задачи много больше значения функции цели исходной задачи. Штрафная добавка $M \sum_{i=1}^{m} \xi_i$ перестает мешать минимизировать функцию цели, только тогда, когда все дополнительные переменные равны 0, т.е. все дополнительные переменные выведены из базиса. Т.о. минимум М задачи будет достигаться в тех же точках, что и минимум исходной задачи, и его численное значение равно численному значению исходной задачи (если исходная задача разрешима). Если не удается привести все вспомогательные переменные к 0, то область определения исходной задачи пуста. Если М задача не разрешима, то и исходная задача неразрешима.

Двойственный симплекс метод.

Пусть ЗЛП задана в виде:

$$f(\vec{x}) = c_0 - \sum_{j=1}^{n} c_j x_j \to max$$

$$\begin{cases} x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j, & i = \overline{1,m} \\ x_j \geq 0, & j = \overline{1,n+m} \end{cases}$$

Здесь свободные члены b_i могут быть как положительными, так и отрицательными. Запишем этот план задачи (возможно и недопустимый) в симплекс таблицу:

	1	X1	X2		Xn
Xn+1	B1	A11	A12		A1n
Xn+2	B2	A21	A22		A2n
Xn+m	Bm	Am1	Am2		Amn
F(x)	C0	C1	C2		Cn

Опишем алгоритм двойственного симплекс метода в виде следующей последовательности действий:

1) В строке функции цели отыскиваем любой коэффициент $c_j < 0, \ j = \overline{1,n}$, пусть это c_k, в этом случае переходим к шагу 2. Если все такие коэффициенты неотрицательны, то в этом случае c_0 будет оценкой целевой функции сверху. При этом мы должны приступить к отысканию опорного решения, которое будет и оптимальным, для этого переходим к шагу 4.

2) В k – м столбце отыскиваем любой положительный коэффициент a_{rk}. Строку r берем в качестве разрешающей. И переходим к 3-му пункту алгоритма.

Если же все коэффициенты в k-м столбце неположительны, то либо задача несовместна, либо линейная форма неограниченна сверху.

3) Вычисляем неположительные отношения коэффициентов строки целевой функции c_j к соответствующим коэффициентам a_{rj} разрешающей строки, и среди этих отношений отыскиваем наибольшее (наименьшее по модулю). Столбец с таким отношением берем в качестве разрешающего(q – й столбец), а элемент a_{rq} берем в качестве генерального. В случае, когда наибольшее отношение равно нулю, его знаменатель a_{rq} берем в качестве разрешающего только в том случае, если он отрицателен (во избежание зацикливания решения). Затем выполняем один шаг модифицированных жордановых исключений и возвращаемся в пункт 1 алгоритма.

4) В качестве разрешающей выбираем любую строку с отрицательным свободным членом b_i (пусть это r – я строка) и переходим к пункту 5 алгоритма.

В случае, если все b_i неотрицательны, то данный опорный план является оптимальным.

5) Берем все отрицательные коэффициенты разрешающей строки a_{rj} и делим на них соответствующие коэффициенты строки целевой функции и находим среди этих отношений большее. Положим это достигается в столбце с номером q, тогда этот столбец берем в качестве разрешающего, а элемент a_{rq} в качестве генерального. Выполняем один шаг модифицированных жордановых исключений и возвращаемся в пункт 4 алгоритма.

В случае, когда в разрешающей строке нет отрицательных коэффициентов a_{rj} , задача несовместна.

Билет 44. Понятие транспортной задачи по критерию стоимости и свойства таких задач.

Постановка транспортной задачи(ТЗ): m – пунктов отправления (ПО) с количеством груза a_i , n – пунктов назначения (ПН) с возможностью принять количество груза b_i , c_{ij} - стоимость перевозки единицы груза из
i – го ПО в j – й ПН. Обозначим за x_{ij} – количество груза перевезенного из i – го ПО в j – й ПН.
Требуется минимизировать стоимость перевозки груза.
Общая математическая модель:

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

$$\begin{cases} \sum_{i=1}^{m} x_{ij} \leq a_i \,, \; i = \overline{1, m} \\ \sum_{j=1}^{n} x_{ij} \geq b_j \,, \; j = \overline{1, n} \\ x_{ij} \geq 0 \,, i = \overline{1, m} \,, j = \overline{1, n} \end{cases}$$

Удобной формой постановки транспортной задачи является сбалансированная форма:

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

$$\begin{cases} \sum_{i=1}^{m} x_{ij} = a_i \,, \; i = \overline{1, m} \\ \sum_{j=1}^{n} x_{ij} = b_j \,, \; j = \overline{1, n} \\ x_{ij} \geq 0 \,, i = \overline{1, m} \,, j = \overline{1, n} \\ \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j \end{cases}$$

Любую несбалансированную ТЗ можно привести к сбалансированной форме.
1) $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$, тогда вводится фиктивный ПН с объемом потребления $\sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j$, удельная стоимости перевозки в который равна 0.
2) $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$, тогда вводится фиктивный ПО с объемом поставки $\sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$, удельная стоимости перевозки из которого во все ПН равна 0.
Транспортную задачу удобно решать используя матрицу перевозок:

	B1		Bn
A1	c11 x11		c1n x1n
Am	cm1 xm1		cmn xmn

Основные свойства ТЗ:
1) Ранг матрицы ограничений сбалансированной ТЗ равен m+n-1, т.е. при решении будет m+n-1 БП.
Всего в задаче m+n ограничений, и условие сбалансированности позволяет выразить одно из ограничений через другие, значит, ранг матрицы ограничений не превосходит m+n-1.
Покажем, что все m+n-1 переменных можно выразить через остальные переменные.

	B1=Σ	B2= Σ		Bn= Σ
A1= Σ	X11	X12		X1n
A2= Σ	X21	X22		X2n
Am= Σ	Xm1	Xm2		Xmn

Переменные из отмеченных ячеек можно выразить через переменные из переменных усеченной матрицы (с индексами i=2…m, j=2…n). Значения Ai и Bj известны, и являются суммами по строкам и по столбцам соответственно. Отмеченных переменных m+n-2. Покажем также, что можно выразить X11, через переменные из усеченной матрицы: это очевидно, поскольку X11 выражается через A1 и прочие элементы 1-й строки, которые в свои очередь выражены через переменные усеченной матрицы. Таким образом показано, что ранг матрицы ограничений по крайней мере равен m+n-1, но в первой части доказательства было показано, что он не может превосходить этого значения, значит ранг системы ограничений в точности равен m+n-1.

2) сбалансированная ТЗ всегда разрешима.

Из ограничений транспортной задачи следует, что $x_{ij} \leq \min_{i,j}\{a_i, b_j\}$ – значит область определения задачи является ограниченным множеством, т.е. в условия ЗЛП – выпуклым многогранником, значит эта задача разрешима.

3) Если a_i и b_j – целые числа, то хотя бы одно из оптимальных решений ТЗ целочисленное.

Билет 45. Циклы в транспортной матрице. Связь между базисными и небазисными переменными в транспортной задаче.

Для ТЗ как и для любой ЗЛП переменные делятся на БП и НБП. m+n-1 БП и m*n-(m+n-1) НБП. Клетки, которые соответствуют БП, называются базисными, другие клетки – небазисные. Значения переменных в небазисных клетках равны 0 (не будем их писать, оставим клетку пустой), в базисные клетки будем вписывать значения соответствующий базисных переменных (даже если они равны 0).

Циклами в матрице ТЗ будем называть ломанные с вершинами, лежащими в клетках матрицы и звеньями, проходящими по строкам и по столбцам матрицы ТЗ. При этом:

1) Ломанная связана (из любой ее вершины можно попасть в любую другую ее вершину, двигаясь в одном и том же направлении).

2) Каждая из вершин цикла смежна с двумя звеньями, причем одно из них проходит вдоль столбца, а другой вдоль строки.

Вершины, являющиеся концами одного звена называются соседними, значит каждая вершина имеет две соседние(одна лежит в том же столбце, другая в той же строке).

Лемма 1:

Если в матрице ТЗ m>=2 и n>=2 отметить m+n клеток, то можно построить по этим клеткам (возможно не по всем) цикл.

Смысл доказательства: если отмеченная клетка единственная в строке или столбце, то вычеркиваем ее из списка отмеченных, так и строим цикл.

Лемма 2:

Число вершин цикла четно. Рассмотрим только звенья лежащие вдоль столбцов, число вершин каждого звена 2 – значит общее число этих звеньев четно, а это и есть все вершины цепи.

Пометим какую-либо клетку цикла символом «+», будем обходить цикл начиная с этой вершины отмечая вершины поочередно знаками «-» и «+», такой цикл будем называть ОЗНАЧЕННЫМ.

Лемма 3:

Построим какой либо означенный цикл в транспортной матрице, количество положительно помеченных вершин в строке (столбце) равно количеству отрицательно помеченных.

Впишем в матрицу ТЗ произвольное решение. Построим по клеткам со значениями цикл, означим его. Возьмем произвольное число Δx, в положительно помеченных клетках, будем добавлять это число к значению в клетке, в отрицательно помеченных будем отнимать. Такую операцию назовем сдвигом по циклу пересчета на величину Δx.

Теорема о сдвиге цикла:

Сдвиг цикла пересчета на какую-либо величину переводит текущее решение ТЗ в другое решение.

По лемме 3, число положительно и отрицательно помеченных вершин по строке и по столбцу одинаково, значит, при сдвиге цикла суммы по строкам и по столбцам будут оставаться теми же.

Лемма 4: для любой транспортной матрицы не существует цикла с вершинами только в базисных клетках.

Положим, что лемма неверна, тогда осуществим сдвиг цикла пересчета из БП: в этом случае значения БП меняются при неизменных значениях НБП, чего не может быть.

Циклом пересчета в ТЗ называется означенный цикл, состоящий из нескольких БП и одной НБП (НБП положительно помечена).

Теорема о цикле пересчета:

Для любой НБП существует цикл пересчета, причем единственный.

Связь между БП и НБП в ТЗ.

Будем решать ТЗ как обычную ЗЛП. Выразим (m+n-1) – переменных через оставшиеся НБП.

$x_{kl} = \beta_{kl} + \sum_{x_{ij}-\text{нбп}} q_{ij}^{(kl)} x_{ij}$ (1)

Приравняем к 0 все НБП, но разрешим изменяться какой-то одной из них. Выпишем остаток от выражения $x_{kl} = \beta_{kl} + q_{ij}^{(kl)} x_{ij}$ (2)

Осуществим сдвиг по циклу пересчета на величину Δx_{ij}, тогда

$x_{kl} = \beta_{kl} + q_{ij}^{(kl)} \Delta x_{ij}$, (3) **коэффициент $q_{ij}^{(kl)}$ в такой задаче может принимать только значения** -1, 0, 1

Причем $q_{ij}^{(kl)} = \begin{cases} 1, \text{если } x_{kl} - \text{положительно помечена} \\ 0, \quad \text{если } x_{kl} - \text{не входит в цикл} \\ -1, \text{если } x_{kl} - \text{отрицательно помечена} \end{cases}$

Осталось посмотреть с какими коэффициентами НБП входит в функцию цели.

$$f(\overline{x}) = \gamma_0 + \sum_{x_{ij} - \text{НБП}} \gamma_{ij} x_{ij}, \text{где } \gamma_{ij} = c_{ij} + \sum_{x_{kl} - \text{БП}} q_{ij}^{(kl)} c_{kl}$$

Фактически γ_{ij} – рассчитывается как алгебраическая сумма удельных стоимостей по циклу пересчета.

46.распределительный метод решения транспортной задачи.

$\{x_{ij}\}$ - множесво НБП опорного плана

$\{x_{k,l}\}$ множество БП опорного плана

$\{\gamma_{i,j}\}$ множество переменных с которым НБП входят в выражение для линейной формы задачи $i=1,m \quad j=1,n \quad k=1,m \quad l=1,n$

Алгоритм симплекс метода для решения задачи минимизации линейной формы:

1) Находим начальный опорный план задачи (методом С-З угла например) и заполняем таблицу $\gamma_{i,j}$ -суммы стоимостей.

2)Для ТЗ строим циклы пересчета всех свободных клеток и находим $\gamma_{i,j}$, при этом учитывая что γ_j берется в симплекс методе с учетом ”-” перед скобками. Т.о. если все $\gamma_{i,j}$ >=0 то наблюдаемый опорный план оптимален и переходим к 6)

(3) линейная форма ТЗ всегда ограничена

4) Выбираем НБП $x_{i,j}$ для которой сумма стоимостей по циклу пересчета $\gamma_{i,j}<0$, строим соответсвующий ей цикл пересчета и среди БП, отрицательно означенных, находим ту значение которой минимально $x_{k,l}$

5) осуществляем сдвиг НБП $x_{i,j}$ на велечину $x_{k,l}$, при этом мы перейдем к новому опорному плану в котором переменная $x_{i,j}$ станет БП а $x_{k,l}$ НБП. Перейти к п.2

6) найденный опорный план оптимальный. выход

Пример из другого источника:

Сущность распределительного метода состоит в том, что для каждой свободной клетки находится цикл, в который входят, кроме неё, только заполненные клетки. С помощью этого цикла определяют, на сколько изменятся транспортные расходы, если ввести в свободную клетку единицу груза. Эта величина k_{ij} называется *индексом свободной клетки* (i, j). Если $k_{ij} < 0$, то в клетку вносится максимально возможная перевозка (она равна минимальной перевозке в «отрицательных» клетках цикла), а если $k_{ij} \geq 0$, то маршрут (i, j) использовать не стоит и проверяется следующая клетка. Процесс заканчивается, когда выясняется, что для всех свободных клеток $k_{ij} \geq 0$.

Оптимальный план следует искать среди планов, в которых заполненные клетки не образуют циклов. Обычно в транспортной задаче число заполненных клеток в точности равно $m + n - 1$, и цикл можно построить единственным образом (например, в рассмотренной задаче число заполненных клеток равно 3 + 4 – 1 = 6). Если включить в цикл свободные клетки со знаком «+», то после изменения плана в нём может оказаться больше $m + n - 1$ заполненных клеток и план будет содержать циклы.

Методы получения первого ДБР ТЗ:

"Метод северо-западного угла (диагональный) ". На каждом этапе максимально возможным числом заполняют левую верхнюю клетку оставшейся части таблицы. Заполнение таким образом, что полностью выносится груз из Ai или полностью удовлетворяется потребность Bj.

"Метод наименьшего элемента ". На каждом этапе заполняют клетку оставшейся части таблицы с наименьшим тарифом.

47.метод потенциалов для решения ТЗ.

Дана задача (1) (2) (3) (4)

(1) $f(\bar{x}) = \sum_{i}^{m}\sum_{j}^{n} C_{i,j}x_{i,j} \to \min$

(2) $\sum_{j=1}^{n} x_{i,j} = a_i \quad i = 1,m \quad \lambda_i$

(3) $\sum_{i=1}^{m} x_{i,j} = b_j \quad j = 1,n \quad \lambda_j$

(4) $x_{i,j} \geq 0 \; i = 1,m \; j = 1,n$

Изменим $f(\bar{x}) = \sum_{i}^{m}\sum_{j}^{n} C_{i,j}x_{i,j} \to \max$ (1*), построим двойственную к ней задачу (5) $f(\bar{\lambda}) = \sum_{i=1}^{m} \lambda_i a_i + \sum_{j=1}^{n} \lambda_j b_j - > \min$ ($m \times n$) огранечений типа (6) $\lambda_i + \lambda_j \geq -C_{i,j} \; i = 1,m, \;\; j = 1,n$, согласно второй теореме двойственности для оптимизационных прямой и двойственной задач выполняется соотношения $(C_{i,j} + \lambda_i{}^* + \lambda_j{}^*)x_{i,j}{}^* = 0$

$u_i = -\lambda_i; \;\; v_j = -\lambda_j$

$(C_{i,j} - (u_i + v_j))x_{i,j}^* = 0$, если $x_{i,j}^* = 0$ то обязательно должно быть выполнено $C_{i,j} - (u_i + v_j) \geq 0$, если $x_{i,j}^* \geq 0$ тогда требуется чтобы $C_{i,j} - (u_i + v_j) = 0$, В любом ДБР сбалансированной ТЗ (n+m-1) БП а остальные НБП, БП те у которых $x_{i,j}^* \geq 0$ тогда для них $C_{i,j} - (u_i + v_j) = 0$ и решая эту систему присваиваем например $u_1 = 0$ получаем однозначно все потенциалы и соответственно $\gamma_{i,j} = C_{i,j} - (u_i + v_j)$.

Алгоритм:

1) находим начальной опорный план ТЗ и вносим его в таблицу перевозок
2) составляем систему и вычисляем потенциалы соответсвующие данному плану.
3) вычисляем $\gamma_{i,j} = C_{i,j} - (u_i + v_j)$ для все НБП если все $\gamma_{i,j} \geq 0$ то перходим у п.6.
4) выбираем НБП для которой $\gamma_{i,j} < 0$, строим ее цикл пересчета и находим БП соответсвующие отрицательно означенным вершинам цикла, ту значение которой минимально $x_{k,l}$
5) производим сдвиг по циклу пересчета НБП $x_{i,j}$ на величину $x_{k,l}$. При этом мы перейдем к новому опорному плану в котором $x_{i,j}$ станет БП а $x_{k,l}$ - НБП перейти к п. 2.
6) Закончить работу, т.к наблюдаемый план является оптимальным.

48. усложненная постановка транспортной задачи. Метод решения транспортной задачи по критерию времени.

1) несбалансированная ТЗ, когда запасы превышают потребости

$$\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$$

$$\begin{cases} f(\bar{x}) = \sum_{i}^{m}\sum_{j}^{n} C_{i,j}x_{i,j} \to \min \\ \sum_{j=1}^{n} x_{i,j} \le a_i \quad i = 1,m \\ \sum_{i=1}^{m} x_{i,j} = b_j \quad j = 1,n \\ x_{i,j} \ge 0 \;\; i = 1,m \;\; j = 1,n \end{cases}$$

Такая задача должна быть приведена к сбалансированному виду и может быть решена методом потенциалов.

Приводим: 1) введем дополнильный фиктивный (n+1) пункт назначения с величиной запасов

$b_{n+1} = \sum a_i - \sum b_j;\; C_{i,n+1} = 0$

2) записываем:

$$f(\bar{x}) = \sum_{i}^{m}\sum_{j}^{n} C_{i,j}x_{i,j} \to \min$$

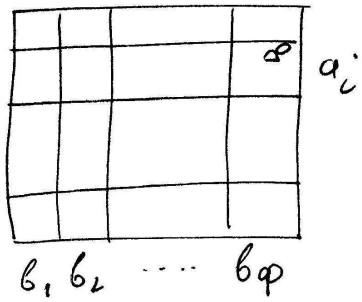
$$\sum_{j=1}^{n+1} x_{i,j} = a_i \quad i = 1,m$$

$$\sum_{i=1}^{m} x_{i,j} = b_j \quad j = 1,n+1$$

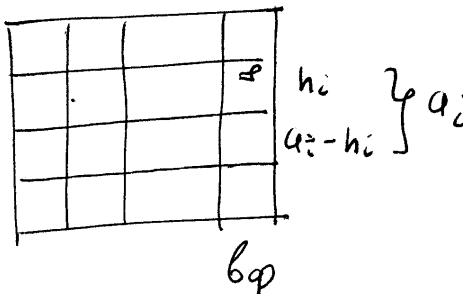
$$x_{i,j} \ge 0 \;\; i = 1,m \;\; j = 1,n+1$$

2) $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$ вводим (m+1) фиктивный пункт отправления с $a_{m+1} = \sum b_j - \sum a_i;\; C_{m+1,j} = 0$

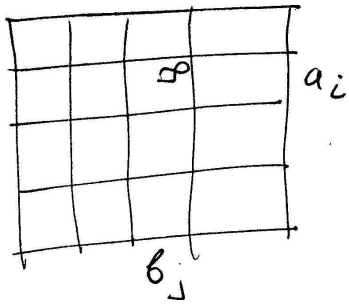
$$\begin{cases} f(\bar{x}) = \sum_{i}^{m}\sum_{j}^{n} C_{i,j}x_{i,j} \to \min \\ \sum_{j=1}^{n} x_{i,j} = a_i \quad i = 1,m \\ \sum_{i=1}^{m} x_{i,j} \le b_j \quad j = 1,n \\ x_{i,j} \ge 0 \;\; i = 1,m \;\; j = 1,n \end{cases}$$



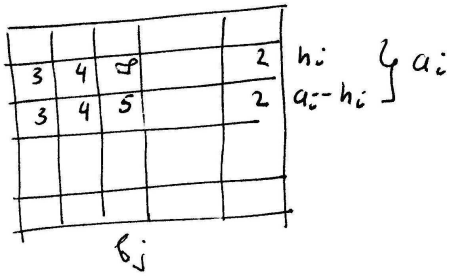
3) $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$ для i ПО весь груз должен быть вывезен реальным потребителем.



4) $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$ для i ПО hi часть запаса должна быть вывезена реальным потребителем.



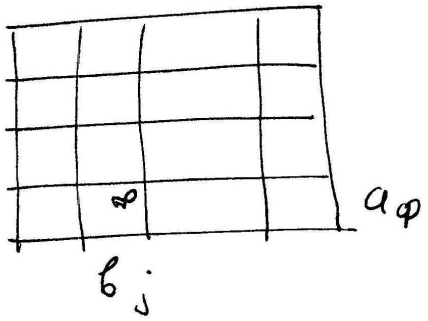
5) ТЗ с запретом некоторого маршрута перевозки (такие задачи могут не името решения)



6) доп условие: по маршруту от i ПО к J ПН величина перевозимого груза ограничена величиной hi

7) $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$ вводим фиктивное ПО

$a_{m+1} = \sum_{j=1}^{m} b_j - \sum_{i=1}^{m} a_j$. Доп условие: j-му потребителю должен быть доставлен реальный груз



8) ТЗ с потерями.

Доп условие: любая единица не доставленного груза приводит к дополнительной потере у потребителя или недовывоз у продавца .

$\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$ введем y_i величина недопоставки j-потребителю, которая приводит к потере на единицу у, r_i тойже размерности $|C_{i,j}|$.

Спланировать перевозки ,минимальную сумму перевозок и потерь. Эту задачу превратить в класическу можно 1) добавив (m+1) поставщика $a_{m+1} = \sum b_j - \sum a_i$ 2) $C_{m+1}, j \overset{def}{=} r_j,\ j = 1,n$

3) $x_{m+1},\ j \overset{def}{=} y_j,\ \ j = 1,n$

$$f(\bar{x}) = \sum_{i}^{m}\sum_{j}^{n} C_{i,j}x_{i,j} + \sum_{j=1}^{n} r_j y_j \to \min \quad \text{у}$$

$$\sum_{j=1}^{n+1} x_{i,j} = a_i \quad i = 1,m$$

$$\sum_{i=1}^{m} x_{i,j} + y_j = b_j \quad j = 1,n+1$$

$$x_{i,j} \geq 0 \ i = 1,m \ \ j = 1,n+1 \quad \text{с}$$

$$y_j \geq 0, \quad j = 1,n$$

$$f(\bar{x}) = \sum_{i}^{m+1}\sum_{j}^{n} C_{i,j}x_{i,j} \to \min$$

$$\sum_{j=1}^{n+1} x_{i,j} = a_i \quad i = 1,m+1$$

$$\sum_{i=1}^{m+1} x_{i,j} = b_j \quad j = 1,n$$

$$x_{i,j} \geq 0 \ i = 1,m+1 \ \ j = 1,n$$

9)многоиндексная ТЗ. Неравенства 4-х индексные, дополнительно к формулировке обычной ТЗ заданы виды груза и виды транспорта . переменные: $x_{i,j,k,l}$ перевозка от i ПО к j ПН k-го вида груза l-м видом транспортаю математическая модель:

$$f(\bar{x}) = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p}\sum_{l=1}^{q} C_{i,j,k,l}x_{i,j,k,l}$$

$$ПО: \sum_{j}^{n}\sum_{l}^{q} x_{i,j,k,l} = a_{i,k} \ \ i = 1,m \ \ k = 1,p$$

$$ПН: \sum_{i}^{m}\sum_{k}^{p} x_{i,j,k,l} = b_{j,k} \ j = 1,n \ \ l = 1,q$$

$$x_{i,j,k,l} \geq 0 i = 1,m \ \ k = 1,p \ \ j = 1,n \ \ l = 1,q$$

10) Распределительные ТЗ. Доп условие: одно еденица доставляемого от i поставщика груза в условиях j-го потребителя закрывает $b_{i,j}$ едениц груза. Надо так распределить перевозки груза , чтобы общая стоимость развозки была наименьшей.

$$\begin{cases} f(\bar{x}) = \sum_{i}^{m}\sum_{j}^{n} C_{i,j}x_{i,j} \to \min \\ \sum_{j=1}^{n} x_{i,j} \leq a_i \quad i = 1,m \\ \sum_{i=1}^{m} x_{i,j}b_{i,j} = b_j \quad j = 1,n \\ x_{i,j} \geq 0 \ i = 1,m \ \ j = 1,n \end{cases}$$

Транспортная задача по критерию времени.

$t_{i,j}$ время доставки всего груза от i ПО к j ПН. Требуется спланировать перевозку чтобы время окончания перевозки всех грузов было наименьшим. Целевая функция нелинейная. Допустим $\bar{x}$ - допустимый план перевозки, оценка эффективности x: $t_{\bar{x}} = \max\limits_{\forall x_{i,j}>0}\{t_{i,j}\}_{\bar{x}}$

$t_{\bar{x}^*} = \min\limits_{\bar{x}} \max\limits_{\forall x_{i,j}>0}\{t_{i,j}\}_{\bar{x}}$, строгий метод решения таких задач на основе симплекс-метода есть.

Алгоритм:

Первый этап (предварительный) : выполняется 1 раз . Построение 1 ДБР исходной задачи, если такое ДБР не удается найти значит исходная задача несовместна иначе переходим к о второму этапу.

Второй этап: имеем ДБР, БП: $x_1, x_2, ..., x_m$ НБП: $x_{m+1}, x_{m+1}, ..., x_n$

1) для текущего набора БП находим величину $t_i = \max\{t_1, t_2, ..t_m\}$
2) просматриваем столбцы с НБП и удаляем те столбцы для которых $t_j \geq t_i$ (найденной). Считаем значения тех переменных =0
3) смотрим i-ю строку, считаем ее порождающей и выполняем одну или несколько итераций симплекс алгоритма, если есть коэффициент $a_{i,j} > 0$ тогда любой столбец ч таким коэффициентом можно считать разрешающим столбцом, для него по обычным правилам симплекс алгоритма находим разрешающую строку и выполняем симплекс алгоритм (работаем с i-й строкой)
   возможно 2 исхода:
   а) переменная x_i станет НБП, т.е мы получим новый план, который не хуже предидущего. Уходим на начало 2 этапа с новым набором БП
   б) x_i не ушла из БП, но все коэффициенты в ее строке $a_{i,j} <= 0$

   б1) $\beta_i > 0$ (*свободный член*) следовательно текущий план улучшить нельзя и его следует принять в качестве оптимального решения исходной задачи.

   Б2) $\beta = 0$ из всего набора столбцов берем те у которых $a_{i,j} < 0$ и переменные для этих столбцов считаем =0, вычеркиваем их, после чего i-ая строчка будет состоять из нулей – вычеркиваем ее. И в начало 2 этапа

49. Основные понятия о графах и сетях. Метод решения задачи о кратчайшем пути.

Говорят, что задан граф $G=(I, U)$, если задано непустое множество вершин I и множество дуг U и каждый элемент $u \in U$ есть упорядоченная пара (i, j) элементов множества I. Здесь i – начало, а j – конец дуги. Геометрически множество I задается точками, а множество U линиями со стрелками, соединяющими эти точки. Граф называется конечным, если оба множества I и U конечны.

Элементами графа являются дуги, вершины, подмножества вершин. Если вершины i и j принадлежат дуге $u = (i, j)$, то говорят, что дуга u инцидентна вершинам i и j, а сами эти вершины инцидентны дуге u. Такие две вершины называются смежными. Вершина, не инцидентная никакой дуге, называется изолированной.

Путем в графе $G=(I, U)$ называется последовательность дуг $(u_1, u_2, \ldots, u_m)$, начало каждой из которых совпадает с концом предыдущей. Конечный путь, у которого начальная вершина совпадает с конечной, называется контуром. Путь, проходящий через все вершины графа и при том только по одному разу, называется гамильтоновым путем. Замкнутый гамильтонов путь (у которого начальная вершина совпадает с конечной) называется гамильтоновым контуром.

Граф называется сильно связным, если любые две его вершины i и j можно соединить путем, идущим из i в j. Графы с такими свойствами называются ориентированными. Если нас не интересует направление дуги $u = (i, j)$, а только соединение двух вершин i и j, по которому можно проходить в любую сторону (от i к j и от j к i), то в отличие от дуги $u = (i, j)$ такое соединение будем называть ребром $u = [i, j]$ и обозначать линией без стрелок.

Если в определениях пути, контура и сильной связности заменить понятие дуги на понятие ребра, то получим определения цепи, цикла и связности. Граф с такими свойствами называется неориентированным.

Один и тот же граф можно представить графически, размещая различным образом его вершины и соединяющие их дуги.

Два графа $G=(I,U)$ и $G^1=(I^1,U^1)$ изоморфны, если между множествами их вершин I и I^1 существует такое взаимно-однозначное соответствие, при котором вершины в одном графе соединены дугами тогда и только тогда, если соединены соответствующие им вершины в другом графе.

Матрицей смежности графа $G=(I,U)$, содержащего n вершин, называется квадратная матрица $A=\|a_{ij}\|_{n\times n}$, где a_{ij} численно равны числу дуг графа, идущих из i - й вершины в j -ю.

Сетью называется граф, элементам которого поставлены в соответствие некоторые параметры, например, транспортных, электрических, информационных объектов. Для общности рассмотрения таких сетей введены функции на вершинах и дугах. Ниже эти функции рассматриваются для транспортной сети.

Пусть каждой вершине конечного графа $G=(I,U)$ поставлено в соответствие некоторое число d_i - интенсивность вершины. Вершины называются источниками, нейтральными и стоками, если соответственно $d_i>0$, $d_i=0$, $d_i<0$, т.е. $I=I^+UI^0UI^-$, где $I^+=\{i\,|\,d_i>0\}$, $I^0=\{i\,|\,d_i=0\}$, $I^-=\{i\,|\,d_i<0\}$.

Потоком в сети называется совокупность величин x_{ij}, $(i,j)\in U$, удовлетворяющих соотношениям

$$\sum_j x_{ij}-\sum_j x_{ji}=d_i\,,\quad i\in I\,, \tag{3.1}$$

$$x_{ij}\geq 0\,,\quad (i,j)\in U\,, \tag{3.2}$$

т.е. поток - это функция, сопоставляющая каждой дуге (i,j) величину x_{ij}.

Для каждой дуги можно определить функцию стоимости, то есть сопоставить c_{ij}, $(i,j)\in U$, - величину удельной стоимости перемещения единицы потока по дуге (i,j). Тогда задача об оптимальном потоке состоит в нахождении потока значений x_{ij} в сети, который оптимизирует линейную форму

$$\sum_{(i,j)\in U} c_{ij}x_{ij}\,, \tag{3.3}$$

когда могут отыскиваться потоки наибольшей и наименьшей стоимости.

Обычно задача линейного программирования (3.1)–(3.3) называется линейной сетевой задачей или задачей о потоке в сети.

Можно ввести функцию $r_{ij} \geq 0$, такую что

$$0 \leq x_{ij} \leq r_{ij}, \quad (i,j) \in U, \tag{3.4}$$

эта функция называется функцией пропускной способности, а r_{ij} – пропускной способностью дуги (i,j). Тогда задача (3.1)–(3.4) называется задачей о потоке в сети с ограниченными пропускными способностями.

Теперь рассмотрим понятие о разрезе в сети. Возьмем граф $G=(I,U)$ с одним источником s и одним стоком t. Определим понятие разреза в сети, отделяющего s от t. Пусть R и $\bar{R}$ – непересекающиеся множества вершин графа: $R \cap \bar{R} = \phi$, $R \cup \bar{R} = I$ и пусть $s \in R$, $t \in \bar{R}$. Тогда разрезом сети, отделяющим s от t, называется множество дуг

$$U(R) = (R, \bar{R}) = \{(i,j) \mid i \in R, j \in \bar{R}\}.$$

Величиной разреза называется сумма пропускных способностей дуг разреза

$$r(R,\bar{R}) = \sum_{i \in R, j \in \bar{R}} r_{ij}.$$

Разрез, имеющий наименьшую пропускную способность $r(R,\bar{R})$, называется минимальным разрезом и обозначается $(R^*, \bar{R}^*)$. Любой разрез запрещает все пути из источника s в сток t. Следовательно, суммарный поток d, текущий из s в t по всем возможным путям в совокупности, не может быть выше пропускной способности любого разреза, в том числе он не может быть выше пропускной способности минимального разреза: $d \leq r(R^*, \bar{R}^*)$. Максимальным называется поток с наибольшим возможным значением d. Последнее называется величиной максимального потока. Тогда задача

$$f(\bar{x}) = d \to \max,$$

$$\sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} d, & i = s \\ -d, & i = t, \\ 0, & i \neq s, i \neq t, i \in I, \end{cases}$$

$$0 \leq x_{ij} \leq r_{ij}, \quad (i,j) \in U$$

называется задачей о максимальном потоке.

Задача о кратчайшем пути:

Эта задача является частным случаем задачи об оптимальном потоке, в которой кроме пропускных способностей дуг присутствует некоторая функция, определенная на всех дугах.

Пусть в конечном графе $G=(I,U)$ каждой дуге поставлено в соответствие число $c(i,j)=c_{ij}$, называемое длиной дуги. Требуется найти путь наименьшей длины, ведущий из некоторой вершины S в другую вершину t. Математически эта задача записывается так (пропускные способности дуг считаются неограниченными):

$$f(\bar{x})=\sum_{(i,j)\in U} c_{ij}x_{ij}\to\min,$$

$$\sum_j x_{ij}-\sum_j x_{ji}=\begin{cases}1, & i=s,\\ 0, & i\neq s,\\ -1, & i=t\end{cases}\quad i\neq t,\ i\in I,$$

$$x_{ij}\geq 0,\quad (i,j)\in U.$$

Для решения этой задачи используется алгоритм Минти, позволяющий связать кратчайшими путями фиксированную вершину сети со всеми остальными вершинами.

Предварительный шаг. Пометим вершину S числом $\varepsilon_s=0$. Далее действуем следующим образом.

Общий шаг. Пусть $A\subset I$ – множество помеченных вершин и для каждой вершины $i\in A$ известна пометка ε_i. Рассмотрим все дуги (i,j), исходящие из вершин множества A и заканчивающиеся на непомеченных вершинах $j\in\bar{A}$. Найдем для каждой дуги $(i,j)\in(A,\bar{A})$ величину характеристики $h_{ij}=\varepsilon_i+c_{ij}$. Выделим дуги, на которых достигается минимум этих характеристик, и отметим числом $\varepsilon_j=\min\limits_{(i,j)\in(A,\bar{A})}\{h_{ij}\}$ те вершины $j\in\bar{A}$, на которых заканчиваются выделенные дуги. Будем помечать таким образом вершины и выделять дуги до тех пор, пока не пометим вершину t. Число ε_t покажет длину кратчайшего пути от S к t.

Завершающий шаг. Сам кратчайший путь определяем, двигаясь от t к S по выделенным дугам в направлении, обратном их ориентации.

Пример. Для заданной на рис 3.6 сети найти кратчайший путь между вершинами I и IO. На ребрах указаны их длины (каждое ребро допускает движение в обоих направлениях). Решение приведено на том же рисунке. Метки у вершин показывают кратчайшее расстояние ε от начальной вершины и номер N общего шага, на котором данная вершина отмечается.

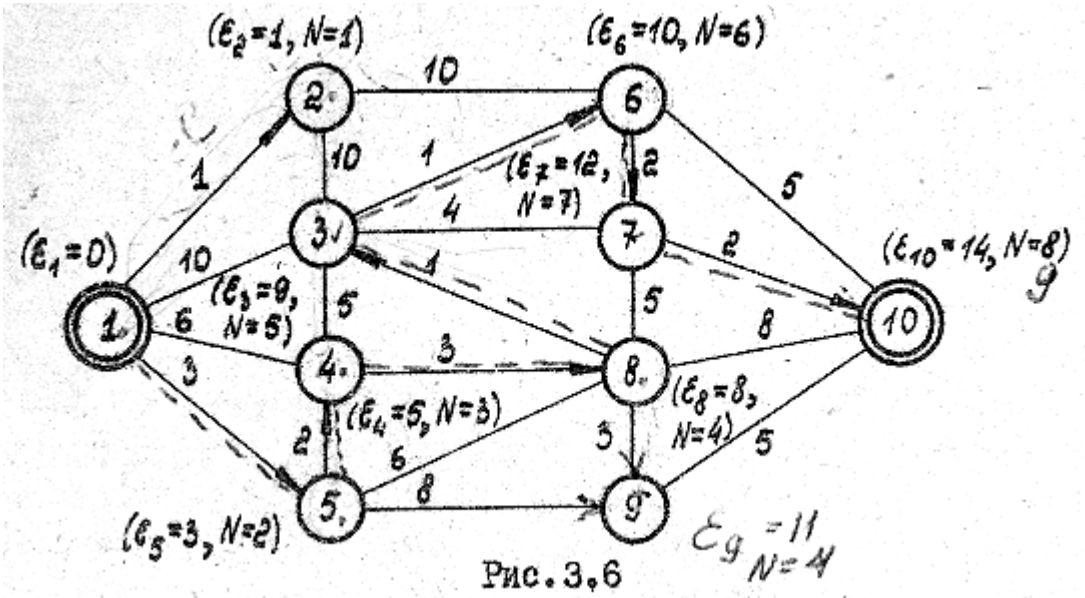

(ε₂=1, N=1)
(ε₆=10, N=6)
(ε₇=12, N=7)
(ε₁=0)
(ε₃=9, N=5)
(ε₄=5, N=3)
(ε₈=8, N=4)
(ε₅=3, N=2)
(ε₁₀=14, N=8)
ε₉=11 N=4
1
2
3
4
5
6
7
8
9
10

Рис. 3.6

50. Метод Форда-Фалкерсона для решения задачи о максимальном потоке сети.

Алгоритм метода основан на следующей теореме.

ТЕОРЕМА. Для любой сети величина максимального потока из истока s в сток t равна пропускной способности минимального разреза, отделяющего s от t.

Таким образом, если величина d суммарного потока равна пропускной способности некоторого разреза сети, то этот поток является максимальным, т.е. задачи о максимальном потоке и о минимальном разрезе являются двойственными: $d_{max} = r(R^*, \bar{R}^*)$.

Доказательство: пусть d – величина максимального потока в сети (I, U). Нужно доказать, что среди всех разрезов сети найдется такой разрез $(R, \bar{R})$, что выполняется: $d = r(R, \bar{R})$.

Оказывается, такой разрез можно построить, если определить множество вершин R рекуррентно так:

$$\begin{cases} s \in R \\ \text{если } i \in R \text{ и } x_{ij} < r_{ij}, \text{ то } j \in R; \\ \text{если } i \in R \text{ и } x_{ji} > 0, \text{ то } j \in R. \end{cases} \qquad (*)$$

Как следует из этого правила, вершины, составляющие множество R, определяются одна за другой, начиная с источника s. Некоторая вершина j включается в R, если существует цепь от s к j, прямые дуги которой не насыщены ($x_{ij} < r_{ij}$), а поток на обратных дугах отличен от нуля. Следовательно, нужно показать, что, во-первых, использование правила $(*)$ приводит к построению разреза, и во-вторых, что для этого разреза имеет место равенство: $d = r(R, \bar{R})$.

Докажем первую часть. Разрез будет построен, если в результате применения правила $(*)$ получим, что $t \in \bar{R}$. Предположим противное, что $t \in R$. Тогда существует цепь из s в t, например $i_1, i_2, \dots, i_m$, где $i_1 = s$ и $i_m = t$. В этой цепи для всех прямых дуг (i_k, i_{k+1}) верно: $x_{i_k, i_{k+1}} < r_{i_k, i_{k+1}}$, а для всех обратных дуг (i_{k+1}, i_k) выполняется: $x_{i_{k+1}, i_k} > 0$. Найдем ε_1 – минимум разности $r - x$, взятой по всем прямым дугам цепи, и ε_2 – минимум x, взятый по всем обратным дугам цепи. Выберем $\varepsilon = \min\{\varepsilon_1, \varepsilon_2\} > 0$. Изменим теперь поток

x_{ij} таким образом: увеличим x_{ij} на величину ε на всех прямых дугах и уменьшим на эту же величину на всех обратных дугах цепи. Величина потока возрастет на ε, а это противоречит максимальности потока d. Значит, $t \in \bar{R}$ и множество дуг $(R, \bar{R})$ есть разрез, отделяющий вершины s и t.

Теперь покажем, что для построенного разреза выполняется: $d = r(R, \bar{R})$. Из правила $(*)$ построения множества R следует, что если $i \in R$ и $j \in \bar{R}$, то должно выполняться: $x_{ij} = r_{ij}$, так как в противном случае вершина j входила бы в множество R. Следовательно

$$\sum_{i \in R,\, j \in \bar{R}} x_{ij} = \sum_{i \in R,\, j \in \bar{R}} r_{ij} = d.$$

Доказанная теорема дает критерий оптимальности потока: следует искать разрез, пропускная способность которого минимальна; эта пропускная способность и будет равна величине максимального потока.

Алгоритм Форда–Фалкерсона

Алгоритм основан на реализации правила $(*)$: начав с некоторого потока (возможно, с нулевого), определяют множество вершин R, которые можно достичь из s по ненасыщенным цепям. Если при этом $t \in R$, то от s к t есть ненасыщенная цепь, и поток по ней можно увеличить; если же $t \in \bar{R}$, то текущий поток максимален, а разрез $(R, \bar{R})$ минимален.

При работе по алгоритму будем на вершинах расставлять пометки. Пометка вершины j может иметь вид (i^+, ε_j) или (i^-, ε_j), где $i \in I$ означает номер предшествующей вершины ненасыщенной цепи от s к t, а ε_j – целое число или ∞ и показывает величину возможного прироста потока от источника s до вершины j. Алгоритм состоит из двух шагов.

Первый шаг. Пометим вершину s меткой $(0, \infty)$; дальнейшая расстановка меток производится в соответствии с правилом $(*)$. Выберем любую помеченную вершину i (первоначально это единственная вершина s). Рассмотрим все непомеченные вершины j, связанные дугами с вершиной i. Тем вершинам j, для которых $x_{ij} < r_{ij}$, припишем метку (i^+, ε_j), где $\varepsilon_j = \min\{\varepsilon_i, r_{ij} - x_{ij}\}$. Тем вершинам j, для которых $x_{ji} > 0$, припишем метку (i^-, ε_j).

где $\varepsilon_j = \min\{\varepsilon_i, x_{ji}\}$. Затем рассмотрим следующую помеченную вершину и поступим аналогично. Этот шаг повторяем до тех пор, пока не пометим вершину t (тогда перейдем ко второму шагу), или пока нельзя будет сделать новых пометок (в этом случае текущий поток максимален).

Второй шаг. Вершина t имеет пометку (j^+, ε_t) или (j^-, ε_t). Если пометка вида (j^+, ε_t), то заменяем x_{jt} на $x_{jt} + \varepsilon_t$, если же пометка вида (j^-, ε_t), то заменяем x_{tj} на $x_{tj} - \varepsilon_t$. Затем переходим к вершине j, в отношении которой поступаем аналогично: если её пометка вида (i^+, ε_j), то заменяем x_{ij} на $x_{ij} + \varepsilon_t$, если же её пометка (i^-, ε_j), то заменяем x_{ji} на $x_{ji} - \varepsilon_t$ и переходим затем всегда к вершине i. Таким образом действуем пока не достигнем источника s. После этого стираем все пометки и снова переходим к первому шагу.

В соответствии с теоремой о максимальном потоке и минимальном разрезе множество дуг, ведущих из помеченных вершин в непомеченные, образует минимальный разрез.

Пример. На рис.3.1 задана сеть; найти максимальный поток из вершины 1 в вершину 6 ; величины r_{ij}, стоящие при ребрах, означают пропускные способности ребер, одинаковые в обоих направлениях.

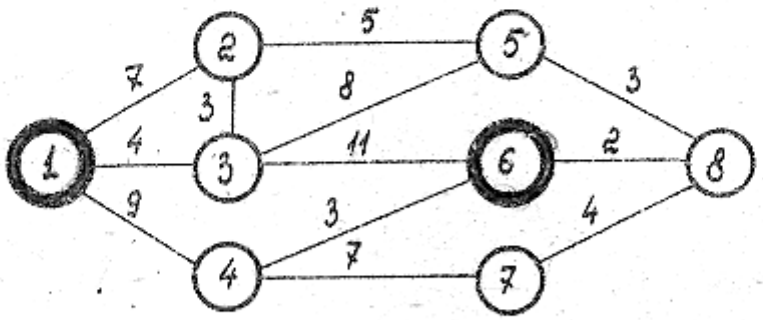


Рис. 3.1

Решение можно начинать с любого потока, в том числе и с нулевого, но решение заканчивается быстрее, если начальный поток ненулевой. В качестве примера на рис. 3.2 построен такой поток. Стрелками показано направление потока. В каждой паре чисел у ребер d обозначает величину потока по дуге, а r – пропускную способность дуги (ребра). Величина начального потока здесь $d_1 = 9$.

На рис.3.2 сформируем пометки как результат первого шага алгоритма.

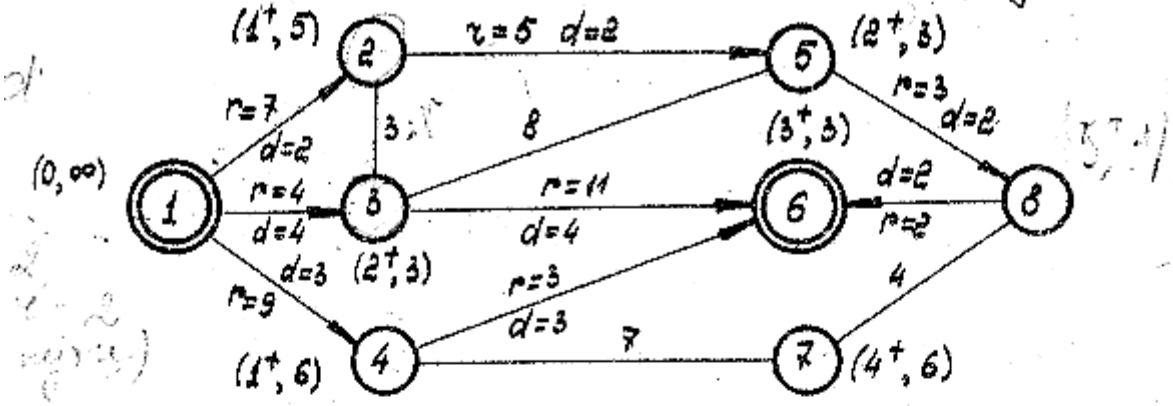


Рис.3.2

Как только окажется помеченным сток, следует перейти ко второму шагу. Результат второго шага показан на рис.3.3, причем $d_1 = 12$.

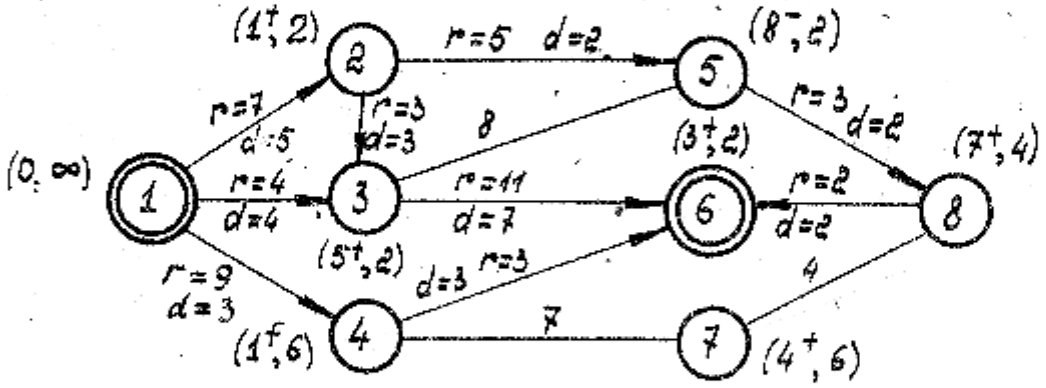


Рис.3.3

Снова применим первый шаг к сети на рис.3.3, соответствующие пометки приведены на том же рисунке. Поскольку помеченной оказалась и вершина 6, применим второй шаг. Его результат показан на рис.3.4, причем $d_1 = 14$. Для сети на рис.3.4 опять выполняем первый шаг (пометки показаны на рисунке). Вершина 6 опять оказалась помеченной. Поэтому переходим к первому шагу. Его результат показан на рис.3.5, причем $d_1 = 16$. Для сети на рис.3.5 снова выполняем первый шаг. Сток пометить не удалось. Следовательно $d_{1\,max} = 16$, и получено оптимальное решение, то есть множества $R = \{1,2,3,4,5,7,8\}$ и $\bar{R} = \{6\}$. Отсюда находим минимальный разрез $(R, \bar{R}) = \{(3,6),(4,6),(8,6)\}$.

Его пропускная способность $r\ (R, \bar{R}) = 11 + 3 + 2 = 16$ равна величине максимального потока.

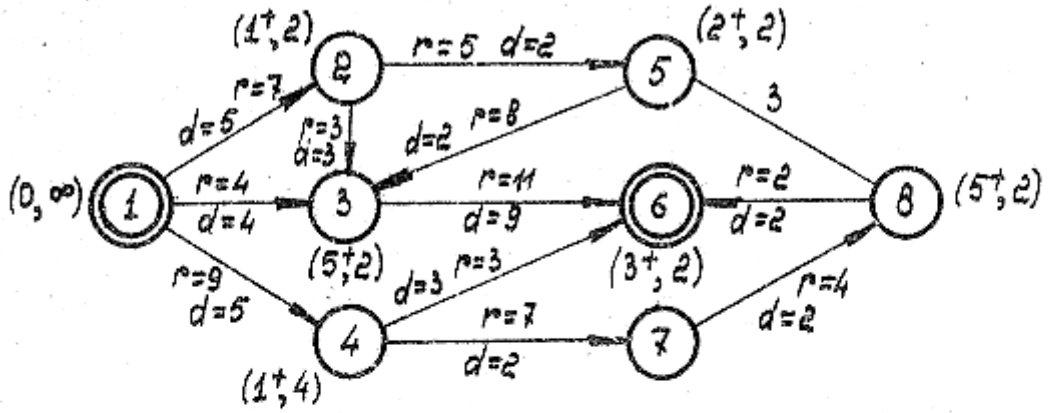


Рис.3.4

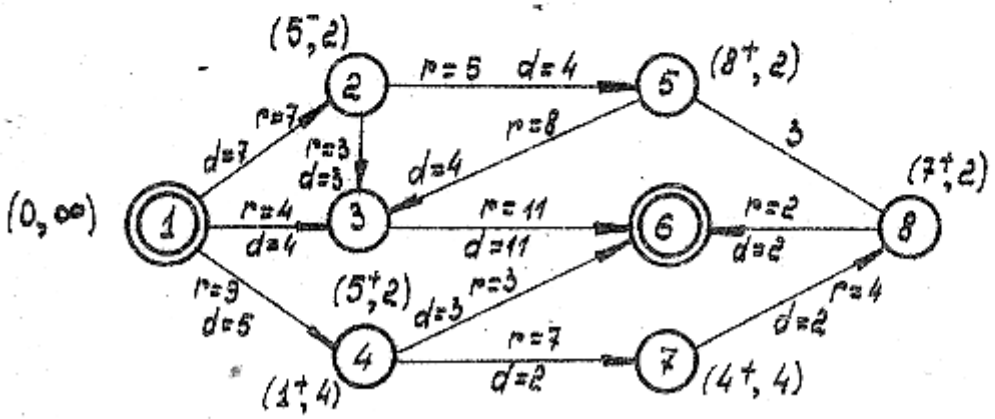


Рис.3.5

51. Линейная сетевая задача, метод потенциалов для ее решения.

Это задача об оптимальном потоке вида

$$f(\bar{x}) = \sum_{(i,j)\in U} c_{ij}\, x_{ij} \to \min,$$

$$\sum_j x_{ij} - \sum_j x_{ji} = d_i, \quad i \in I, \tag{3.5}$$

$$x_{ij} \geq 0, \quad (i,j) \in U.$$

При $\sum_i d_i = 0$, что является необходимым условием баланса для сети, мы имеем сбалансированную транспортную задачу, в которой $d_i > 0$ соответствуют пунктам производства, $d_i < 0$ – пунктам потребления, $d_i = 0$ – промежуточным пунктам, где груз не производится и не потребляется, а только проходит через них, например перегружается. Пропускные способности дуг являются неограниченными.

Согласно второй теореме двойственности, построив двойственную задачу с переменными $u_1, u_2, \ldots, u_N$, где $N = |I|$ – кардинальное число множества I, т.е. число вершин в множестве I, получим следующий признак оптимальности для прямой задачи:

при $x_{ij}^* > 0$ должно выполняться условие $u_j^* - u_i^* = c_{ij}$,

при $x_{ij}^* = 0$ должно выполняться условие $u_j^* - u_i^* \leq c_{ij}$.

Продолжение на следующей странице.

Здесь u_i , v_j – потенциалы. Данный критерий оптимальности совпадает с критерием оптимальности для метода потенциалов при решении транспортных задач, заданных с помощью транспортной матрицы.

Аналогично транспортным задачам в матричной постановке и для сетевых задач легко показать, что ранг системы ограничений (3.5) равен $N-1$. Таким образом, и число базисных дуг на транспортной сети должно быть на единицу меньше числа вершин сети. При этом базисные дуги должны образовать на сети дерево, соединенное со всеми вершинами. Напомним, что граф называется деревом, если он связен (сильно связен) и не имеет циклов (контуров). Наличие цикла указывает на линейную зависимость базисных векторов транспортной задачи.

Алгоритм метода потенциалов для линейной сетевой задачи рассмотрим непосредственно на примере транспортной задачи, заданной на рис.3.7 в виде неориентированного графа с параметрами.

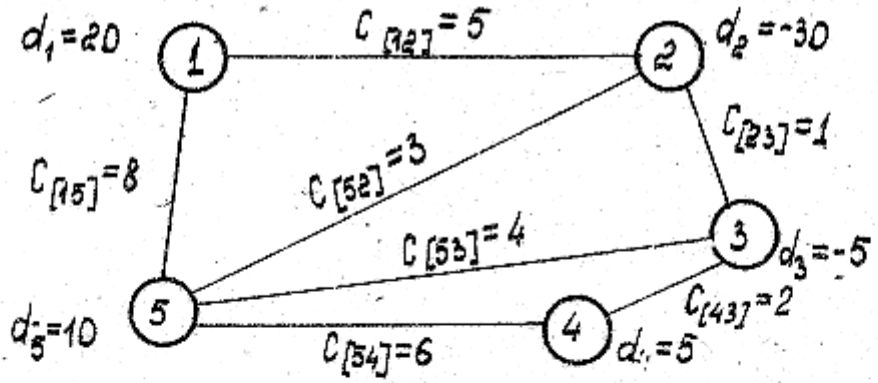


Рис. 3.7

Вершины I,4,5 являются пунктами производства, а вершины 2 и 3 – пунктами потребления. Рядом с каждой вершиной записана величина её запаса (потребности). Каждому ребру соответствует свое значение удельной стоимости, одинаковой для перевозки груза по ребру в любом направлении. Поставки однородного груза из одной вершины в другую будут изображаться дугами с указанием величины поставки. Дуга, выходящая из вершины, означает вывоз груза из этой вершины, а входящая дуга – ввоз груза в эту вершину.

Решение задачи складывается из предварительного и общего

этапов. На первом этапе специальным методом строится допустимое базисное решение сбалансированной сетевой транспортной задачи. На втором этапе с помощью метода потенциалов это допустимое базисное решение оптимизируется.

Предварительный этап. Для построения базионого плана воспользуемся алгоритмом, предложенным М.А.Яковлевой. Он реализуетоя двумя последовательными процессами.

Процесс I. Предназначен для построения системы базионых дуг (т.е. для построения дерева, состоящего из базисных дуг). На данном шаге алгоритма каждую базисную дугу будем отмечать линией с двумя стрелками, направленными в обе стороны. Процесс построения сопровождается расстановкой и корректировкой пометок у вершин. Пометка показывает число базисных дуг, соединенных с вершиной. Построение заключается в следующем (рис.3.8).

Вначале всем вершинам присваивают нулевые пометки. В качестве базисной дуги выбирают любое ребро; мы начнем с ребра [I,5] и пометим его стрелками. Одновременно с этим увеличим на единицу пометки у вершин, связанных этой базисной дугой. Затем поочередно связываем базисными дугами вершины, имеющие нулевые пометки, и вершины с пометками, отличными от нуля. При проведении очередной базисной дуги пометки вершин, которые она соединяет, должны быть увеличены на единицу. Построение закончено, когда все вершины сети будут помечены ненулевыми оценками. На рис.3.8 построение проводилось в следующем порядке: [I,5], [5,3], [5,4], [3,2].

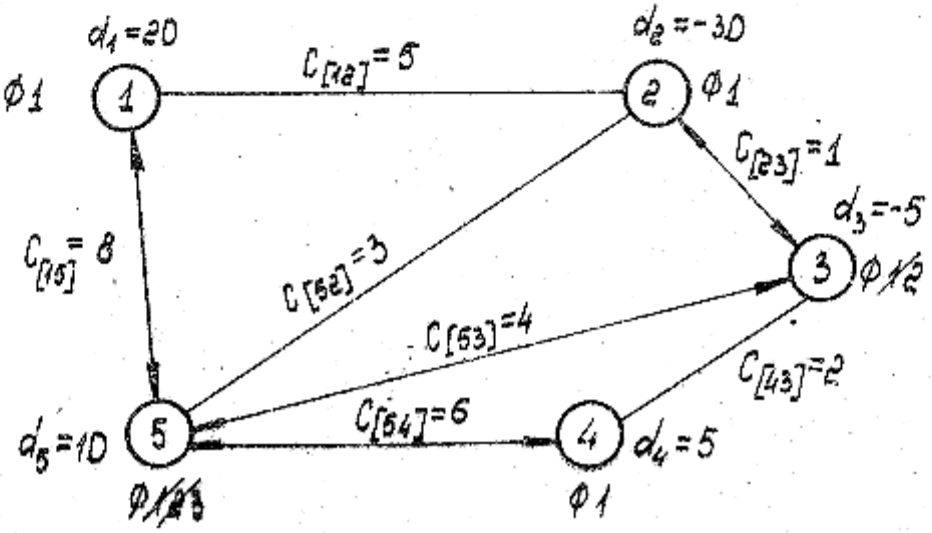


Рис.3.8

Процесс 2. Предназначен для определения величины потока по каждой базисной дуге и однозначной ориентации базисных дуг. Построение осуществляется в следующем порядке (рис.3.9).

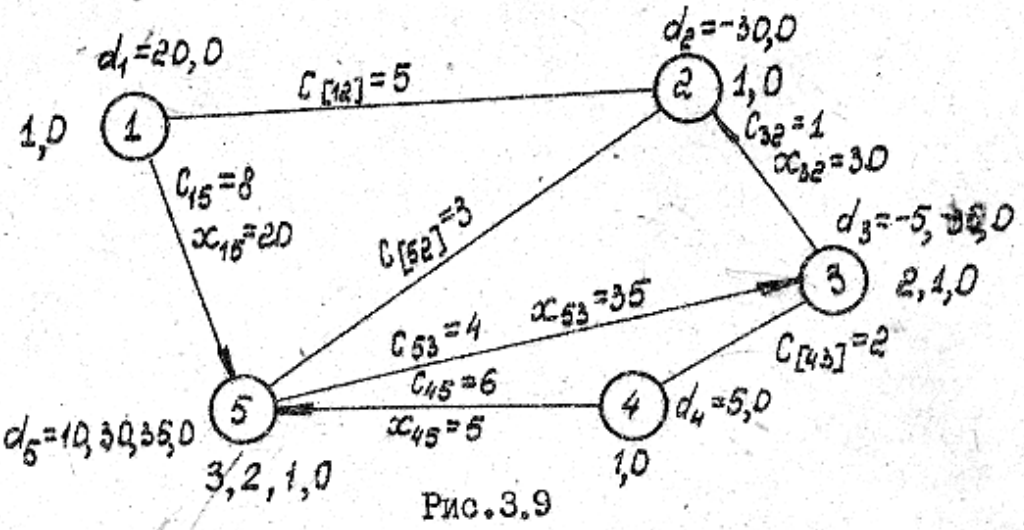


Рис.3.9

Выбираем любую вершину с пометкой - единица. Базисной дуге, примыкающей к вершине, придаем направление, соответствующее знаку d_i (для минуса дуга входит в вершину, для плюса - выходит; если $d_i = 0$ - любое направление дуги). Величину потока по рассматриваемой дуге принимаем равной запасу или потребности груза в вершине. Пометки вершин, соединенных этой дугой, уменьшаем на единицу, а интенсивности у вершин корректируем с учетом потока по дуге. Опять рассматриваем вершину с единичной пометкой и все действия повторяем. Процесс построения закончен, когда пометки у всех вершин - нулевые. В нашем случае мы последовательно образовывали базисные дуги (I,5),(4,5),(3,2),(5,3).

Общий этап. Шаг I. Полученный базисный план должен быть проверен на оптимальность с помощью потенциалов, которые определяются так. Произвольно выбранной вершине присваиваем любое значение в качестве потенциала. Двигаясь по базисным дугам, вычисляем потенциалы всех остальных вершин. Если дуга выходит из вершины, то к потенциалу вершины прибавляем величину C_{ij} удельной стоимости для этой дуги. Если дуга входит, то C_{ij} вычитается из потенциала. На рис.3.10 приведены значения вычисленных потенциалов, причем построение начинали с потенциала U_1 первой вершины, которому присвоено значение 10 единиц.

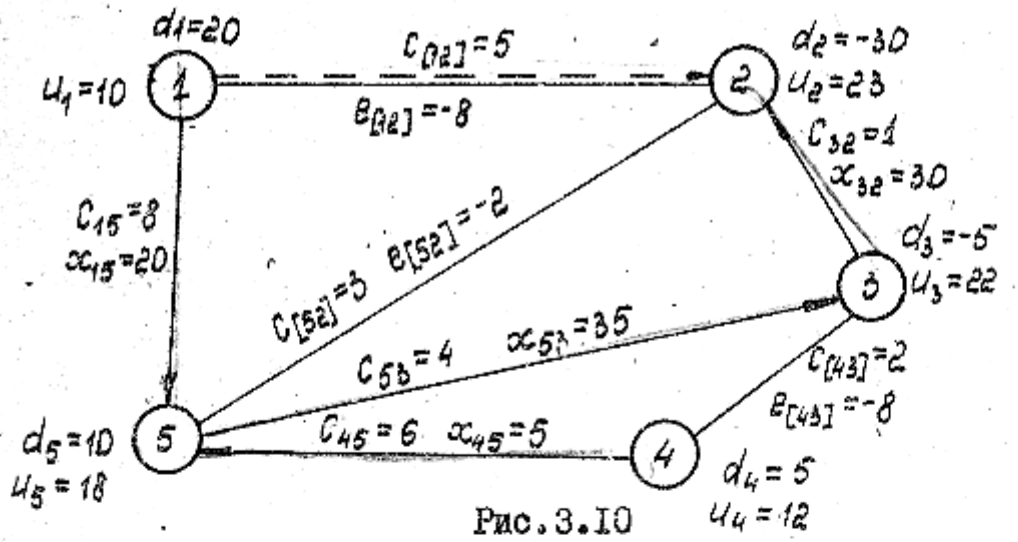


Рис.3.10

Шаг 2. Для определения оптимальности плана определим характеристики e_{ij} для всех небазисных ребер. Каждому ребру инцидентны два потенциала смежных вершин u_i и u_j. Тогда e_{ij} вычисляется как $e_{ij} = c_{ij} - |u_i - u_j|$. Задача считается решенной, если план оптимальный, т.е. характеристики всех небазисных ребер неотрицательны. Рассматриваемый план (рис.3.10)–неоптимальный: $e_{[12]} = -8$, $e_{[52]} = -2$, $e_{[43]} = -8$. Решение продолжается дальше.

Шаг 3. Одно небазисное ребро с отрицательной характеристикой, например $[1,2]$, введем в базис. Составим цикл, состоящий из этого ребра и базисных дуг; такой цикл – единственный и имеет вид: $[1,2]$, $(3,2)$, $(5,3)$, $(1,5)$.

Шаг 4. Ориентируем ребро $[1,2]$ в построенном цикле в направлении от вершины с меньшим потенциалом к вершине с большим потенциалом, в нашем случае к вершине 2. Рассмотрим цепь из базисных дуг, соединяющую конец дуги (1,2) с её началом, т.е. будем двигаться по этой цепи в направлении от вершины 2 к вершине 1. Среди всех базисных дуг, направление которых противоположно нашему движению (т.е. противоположно направлению новой дуги (1,2)), выбираем базисную дугу с наименьшим значением потока на ней. На рис.3.10 среди дуг противоположного направления, т.е. дуг (3,2),(5,3),(1,5), выбираем дугу (1,5) с $x_{15} = 20$. Выполняем перераспределение потоков (поставок груза) в дугах рассматриваемого цикла. Для этого найденная величина потока (20 единиц) прибавляется ко всем потокам в дугах цикла, совпадающих по направлению с новой дугой, и вычитается из потоков в дугах цикла противоположного направления. После этого

выбранная дуга (I,5) с наименьшим значением потока становится небазисным ребром (т.е. на ней ликвидируется стрелка), а новая дуга (I,2) с величиной поставки в ней $x_{12} = 20$ включается в число базисных дуг. При этом общее число базисных дуг в новом плане по-прежнему на единицу меньше числа вершин сети. Новый базисный план, полученный в результате перераспределения потоков, показан на рис.3.II.

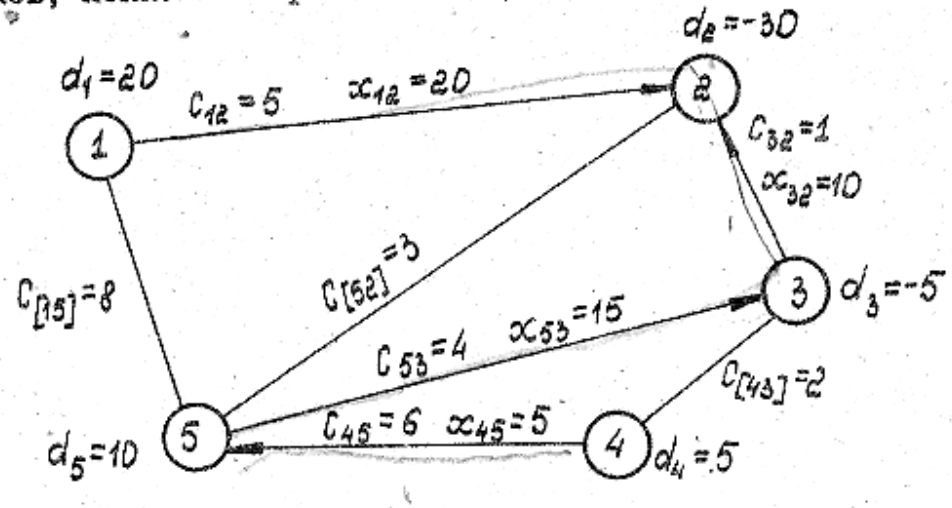


Рис.3.II

Опять приступаем к проверке полученного базисного плана на оптимальность, т.е. переходим к первому шагу общего этапа. Указанная последовательность действий для рассматриваемой задачи привела к уменьшению значения функции цели за одну итерацию. Так, для базисного плана на рис. 3.9

$$f(\bar{x}) = c_{15}x_{15} + c_{45}x_{45} + c_{53}x_{53} + c_{32}x_{32} = 360,$$

а для нового базисного плана на рис.3.II

$$f(\bar{x}) = c_{12}x_{12} + c_{45}x_{45} + c_{53}x_{53} + c_{32}x_{32} = 200.$$

52. Жордановы исключения. Геометрический метод решения задач линейного программирования.

53. Задачи оптимального управления. Принцип оптимальности динамического программирования.

Динамическое программирование – это математический метод оптимизации многошаговых процессов принятия решений, т.е. таких процессов, которые разбиваются на ряд последовательных шагов, распределенных во времени или в пространстве.

В единих задачах этапи или шаги естественны, в других задачах этапы или шаги вводятся в постановку задачи искусственно для того, чтобы можно было воспользоваться методом динамического программирования (например, в постановку задач, связанных с нахождением оптимального размера капиталовложений).

В многошаговых процессах принятия решений любой шаг может быть охарактеризован значениями некоторой функции эффективности, введенной для отдельного шага. Решение или управление, применяемые на данном шаге, влияют на решения, принимаемые на последующих шагах. Эта связь решений обусловливается некоторым общим ресурсом, используемым в системе, связью состояний и управлений на отдельных шагах. Связь шагов идет от природы задач.

Нас интересует не оптимизация любого отдельного шага, а получение оптимального для системы в целом решения, т.е. решения, охватывающего все этапы задачи.

Методы динамического программирования используются для решения задач:

календарного планирования;

распределения ресурсов или капиталовложений между различными областями их использования;

обеспечения надежности сложных технических систем;

определения оптимальных объемов запасных частей, необходимых для эксплуатации сложного оборудования;

замены и ремонта оборудования;

определения оптимальных маршрутов на сетях;

управления запасами и т.д.

Пусть имеется некоторая физическая система S, которая с течением времени меняет свое состояние, т.е. в системе S происходит какой-то процесс. Мы можем управлять этим процессом, т.е. тем или другим способом влиять на состояние системы. Такую систему S мы

будем называть управляемой, а способ нашего воздействия на нее – управлением U. Напомним, что буквой U обозначается не какая-то одна величина, а целая совокупность величин, векторов или функций, характеризующих управление.

Предположим, что с процессом связана какая-то наша заинтересованность, выражающаяся численно величиной F, которую мы будем называть "выигрышем". Мы хотим управлять процессом, чтобы выигрыш был максимален. Здесь и в дальнейшем для краткости будем говорить только о максимизации F; подразумевается, что "максимум" в любом случае может быть заменен на "минимум".

Очевидно, выигрыш зависит от управления:

$$F = F(U).$$

Мы хотим найти такое управление (оптимальное)

$$U = (u),$$

при котором выигрыш максимален:

$$F_{max} = \max \{F(U)\}.$$

Обычно в таких задачах должны быть учтены некоторые условия, накладываемые на начальное состояние системы S_0 и конечное состояние S_ω.

В простейших случаях эти состояния могут быть полностью заданы. В других случаях они могут быть заданы не полностью, а только ограничены какими-то условиями, т.е. указаны область начальных состояний $\hat{S}_0$ и область конечных состояний $\tilde{S}_\omega$.

Таким образом, общая задача оптимального управления формулируется следующим образом.

Из множества возможных управлений U найти такое оптимальное управление, которое переводит физическую систему S из начального состояния S_0 в конечное состояние S_ω так, чтобы при этом выигрыш F обращался в максимум.

Дадим процессу управления геометрическую интерпретацию. Для этого введем понятие о так называемом фазовом пространстве (или пространстве состояний).

Состояние s системы S, которой мы управляем, всегда можно описать с помощью того или другого количества численных параметров. Такими параметрами могут быть, например: координаты тела и его скорость; количества средств, вложенных в отрасль производства и т.д.

Эти параметры мы будем называть фазовыми координатами системы S, а состояние системы изображать точкой s с этими координатами в некотором условном фазовом пространстве (пространстве состояний). Размерность этого пространства зависит от числа фазовых координат.

Если состояние системы характеризуется двумя параметрами ξ_1 и ξ_2, то фазовое пространство будет двухмерным (плоскость или ее часть), а процесс будет изображаться перемещением точки из S_0 в S_w по определенной траектории на фазовой плоскости ξ_1 ξ_2. Траектория эта и будет изображать управление (рис. I.I).

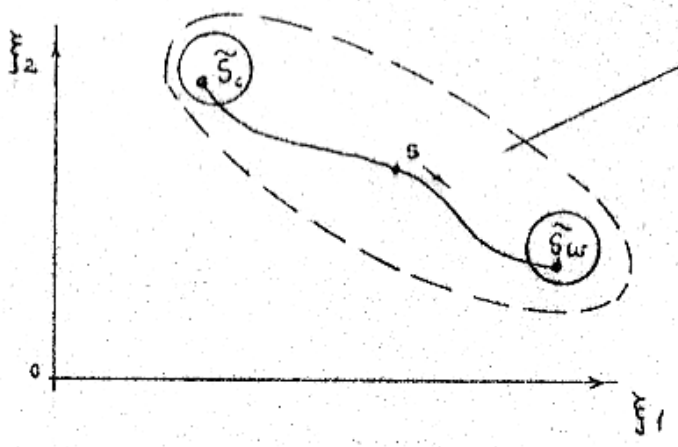


Рис. I.I

Если число параметров, характеризующих состояние системы больше трех, то геометрическая интерпретация теряет свою наглядность, но геометрическая терминология продолжает оставаться удобной. В общем случае, когда состояние системы S описывается n параметрами

$$\xi_1, \xi_2, \dots, \xi_n,$$

мы будем говорить о точке s в n -мерном фазовом пространстве и о ее перемещении из области $\tilde{S}_0$ в область $\tilde{S}_w$ по такой траектории, для которой выигрыш F максимален.

Выбор фазовых координат $\xi_1, \xi_2, \dots \xi_n$, определяющих состояние системы, и соответствующей геометрической интерпретации может быть тем или другим, в зависимости от удобства построения расчетной схемы.

Предположим, что фазовые координаты $\xi_1, \xi_2, \ldots \xi_n$, определяющие состояние системы S, выбраны. Общая задача оптимизации управления в геометрических терминах может быть сформулирована так.

Найти такое управление (оптимальное управление), под влиянием которого точка s фазового пространства переместится из начальной области $\widetilde{S}_0$ в конечную область $\widetilde{S}_\omega$ так, что при этом выигрыш обратится в максимум.

Поставленную общую задачу можно решать различными способами, отнюдь не только методом динамического программирования. Характерным для динамического программирования является определенный математический прием, состоящий в следующем: процесс перемещения точки s из $\widetilde{S}_0$ в $\widetilde{S}_\omega$ разделяется на несколько шагов (этапов), и затем проводится пошаговая оптимизация управления и выигрыша.

Процедура построения оптимального управления методом динамического программирования распадается на две стадии: предварительную и окончательную. На предварительной стадии определяются для каждого шага условное оптимальное управление, зависящее от состояния s системы (достигнутого в результате предыдущих шагов), и условный оптимальный выигрыш на всех оставшихся шагах, начиная с данного, также зависящий от состояния s.

На окончательной стадии определяется для каждого шага окончательное (безусловное) оптимальное управление.

Предварительная (условная) оптимизация производится по шагам в обратном порядке: от последнего шага к первому; окончательная (безусловная) оптимизация – также по шагам, но в естественном порядке: от первого шага к последнему.

Из двух стадий оптимизации несравненно более важной и трудоемкой является первая.

В основе поэтапной процедуры лежит уже упоминавшийся принцип оптимальности, состоящий в следующем.

Каково бы ни было состояние s системы в результате какого-то числа шагов, мы должны выбирать управление на ближайшем шаге так, чтобы оно в совокупности с оптимальным управлением на всех последующих шагах приводило к максимальному выигрышу на всех оставшихся шагах, включая данный.

Запишем принципиальную структуру обеих стадий оптимизации с помощью общих символических формул. "Символическими" мы их называем потому, что в них будут фигурировать функции, аргументами кото-

рых будут не числа, а "состояния" и "управления", каждое из которых в общем случае характеризуется не одним числом, а целой совокупностью чисел или функций.

Обозначим $F(S)$ как условный оптимальный выигрыш, получаемый на всех последующих шагах, начиная с i-го и до конца. Он достигается при оптимальном управлении на всех этих шагах и равен максимальному выигрышу, который можно получить на всех этих шагах вместе, если в их начале система находится в состоянии ; $U_i(S)$ – условное оптимальное управление на i-м шаге, которое совместно с оптимальным управлением на всех последующих шагах обращает выигрыш на всех оставшихся шагах, начиная с данного, в максимум.

Найдем функции $F_i(S)$ и $U_i(S)$ для всех шагов ($i = 1, 2, \ldots, m$).

Для этого рассмотрим i-й шаг процесса управления. Пусть в результате $i-1$ предыдущих шагов система пришла в состояние , и мы выбираем какое-то управление U_i на i-м шаге. Если мы его применим, то получим на данном i-м шаге какой-то выигрыш f_i ; он зависит как от состояния системы S , так и от примененного управления U_i :

$$f_i = f_i(S, U_i).$$

Кроме того, мы получим какой-то выигрыш на всех оставшихся шагах. Соответственно принципу оптимальности будем считать, что он максимален. Чтобы найти этот выигрыш, мы должны знать состояние системы перед следующим, ($i+1$)-м шагом. Под влиянием управления U_i на i-м шаге система из состояния s (в котором она была перед этим шагом) перейдет в какое-то новое состояние S' . Это новое состояние будет зависеть опять-таки от прежнего состояния и примененного управления U_i :

$$S' = \psi_i(S, U_i).$$

Запишем выигрыш, который мы получим на всех шагах, начиная с i-го, если на i-м шаге будет применено любое управление U_i , а на всех последующих от ($i+1$)-го до m-го-оптимальное управление. Этот выигрыш будет равен выигрышу f_i на данном, i-м шаге, плюс условный оптимальный выигрыш на всех последующих шагах, начиная с ($i+1$)-го, определяемый для нового состояния системы S' ; обозначим такой "полуоптимальный" выигрыш через $\tilde{F}(S, U_i)$:

$\tilde{F}_i(S, U_i) = f_i(S, U_i) + F_{i+1}(S')$, $\tilde{F}_i(S, U_i) = f_i(S, U_i) + F_{i+1}(\psi_i(S, U_i))$.

$$F_i(S) = \max_{U_i} \{ f_i(S,U_i) + F_{i+1}(\Psi_i(S,U_i)) \}. \qquad (I.I)$$

То управление

$$U_i = U_i(S),$$

при котором этот максимум достигается, и есть условное оптимальное управление на i-м шаге, а сама величина – условный оптимальный выигрыш (на всех шагах, начиная с i-го, и до конца).

В уравнении (I.I) функции $f_i(S,U_i)$ и $\Psi_i(S,U_i)$ известны. Неизвестными остаются функции $F_i(S)$ и $F_{i+1}(S)$; из них первая выражается через вторую.

Формула (I.I) представляет собой так называемое основное функциональное уравнение динамического программирования; она позволяет определить функцию $F_i(S)$, если известна следующая за ней по порядку функция $F_{i+1}(S)$.

Что касается функции $F_m(S)$ (условный оптимальный выигрыш на последнем шаге), то она может быть определена очень просто. Действительно, за последним шагом нет никакого другого, и нужно попросту обратить в максимум выигрыш на последнем шаге:

$$F_m(S) = \max_{U_m} \{ f_m(S,U_m) \}. \qquad (I.2)$$

Максимум в формуле (I.2) берется не по всем возможным управлениям U_m на m-м шаге, а только по тем, которые приводят систему в заданную область конечных состояний $\tilde{S}_m$, т.е. по тем, для которых

$$\Psi_m(S,U_m) \in \tilde{S}_m.$$

То управление $U_m = U_m(S)$, при котором достигается максимум выигрыша (I.2), и есть условное оптимальное управление на последнем шаге.

Теперь можно одно за другим построить всю цепочку условных оптимальных управлений. Действительно, зная $F_m(S)$, можно по общей формуле (I.I), полагая в ней $i + I = m$, найти функцию $F_{m-1}(S)$ и соответствующее условное оптимальное управление $U_{m-1}(S)$; затем $F_{m-2}(S)$ и $U_{m-2}(S)$ и так далее, вплоть до первого шага, у которого будут найдены функции $F_1(S)$ и $U_1(S)$. Функция $F_1(S)$ есть условный оптимальный выигрыш за всю операцию, т.е. на всех шагах, начиная с первого и до последнего (если первый

шаг начинается с определенного состояния s системы S).

Таким образом, предварительная оптимизация закончена – найдены условный оптимальный выигрыш и условное оптимальное управление для каждого шага.

Теперь перейдем ко второй стадии оптимизации – нахождению безусловного (окончательного) оптимального управления

$$U = (U_1, U_2, \ldots, U_m).$$

Начнем с первого шага. Предположим, что исходное состояние S_0 нам полностью известно. Подставим это состояние S_0 в формулу для условного оптимального выигрыша $F_1(s)$. Получим

$$F_{max} = F_1(S_0). \qquad (I.3)$$

Оптимальное управление на первом шаге найдется одновременно с величиной (I.3):

$$U_1 = U_1(S_0).$$

Далее, зная исходное состояние S_0 и управление U_1 , можем найти состояние S_1^* системы после первого шага:

$$S_1^* = \Psi_1(S_0, U_1).$$

Зная это состояние S_1^* , можно найти оптимальное управление на втором шаге $U_2 = U_2(S_1^*)$, затем $S_2^* = \Psi_2(S_1^*, U_2)$ и т.д. Таким образом, идя по цепочке

$$S_0 \to U_1(S_0) \to S_1^* \to U_2(S_1^*) \to \ldots \to S_{m-1}^* \to U_m(S_{m-1}^*) \to S_m^*,$$

мы определим одно за другим все шаговые оптимальные управления и найдем состоящее из них оптимальное управление операцией в целом:

$$U = (U_1, U_2, \ldots, U_m),$$

а также (если оно не было в точности задано заранее) конечное состояние системы:

$$S_w = S_m^*.$$

Разумеется, это состояние будет принадлежать области S_w^* , потому что мы выбрали управление на последнем шаге именно так, чтобы это условие было соблюдено:

$$S_m^* = \Psi(S_{m-1}^*, U_m) \in \tilde{S}_w.$$

Предположим теперь, что исходное состояние системы известно нам не полностью, а только ограничено условием:

$$S_0 \in \widetilde{S}_0.$$

Тогда нужно найти такое начальное состояние S_0^*, при котором условный оптимальный выигрыш за все шаги максимален:

$$F_{max} = \max_{S \in \widetilde{S}_0} \{F_1(S)\}.$$

То начальное состояние S_0^*, для которого этот максимум достигается, и должно быть выбрано в качестве исходного. Далее оптимальное управление строится совершенно так же, как и раньше, по цепочке:

$$S_0^* \to U_1(S_0^*) \to S_1^* \to U_2(S_1^*) \to \dots \to S_{m-1}^* \to U_m(S_{m-1}^*) \to S_m^*,$$

что и дает оптимальное управление операцией в целом:

$$U = (U_1, U_2, \dots, U_m),$$

и конечное состояние системы $S_\omega = S_\omega^*$, если оно заранее не было полностью определено.

На этом процесс оптимизации заканчивается.

В заключение отметим следующее. В принципе процесс динамического программирования может разворачиваться (хотя и не так естественно) и в направлении, обратном тому, которое мы приняли: условные оптимальные управления могут отыскиваться в направлении от первого шага к последнему, а безусловные – от последнего к первому. Вычислительно эта схема ничуть не хуже предложенной выше, но в смысле удобства изложения и понимания уступает ей. Поэтому в практических приложениях данный метод не получил широкого распространения.

54. Метод динамического программирования для дискретных систем.

Рассмотрим формализованное представление принципа оптимальности Беллмана для дискретных систем (объектов). Вообще-то, основная интерпретация применения динамического программирования относится именно к исследованию дискретно-непрерывных систем или чисто дискретных систем.

Будем рассматривать принцип оптимальности на примере простого объекта, движение которого описывается уравнением первого порядка

$$\frac{dx}{dt} = f_1(x,U),$$

где x является единственной координатой, а U – единственное управляющее воздействие. Таким образом, в данной задаче x и U являются скалярными, а не векторными переменными. Начальное состояние объекта задается условием $x|_{t_0=0} = x^{(0)}$. Пусть необходимо найти управление $U(t)$, минимизирующее функционал

$$F = \int_{t_0}^{T} Q_1(x,U)dt + \Psi_1(x(T)),$$

где t_0 будем считать равным нулю, а величину T для простоты примем фиксированной. Здесь подразумевается, что $U(t) \in \Omega(U)$, например $|U(t)| \le U$.

Проведем дискретизацию этой задачи, т.е. приближенно опишем непрерывную задачу в виде дискретно-непрерывной задачи. Подобный переход часто приходится выполнять и при решении многих практических задач. Прежде всего разобьем интервал $(0,T)$ на N одинаковых участков длиной Δ. В дальнейшем будем рассматривать только дискретные значения $x = x(k)$, $U = U(k)$ при $k = 0, 1, \ldots, N$, для которых $t = 0, 1\Delta, 2\Delta, \ldots, (N-1)\Delta, N\Delta = T$. Представим дифференциальное уравнение в виде разностного:

$$\frac{x(k+1)-x(k)}{\Delta} = f(x(k), U(k))$$

или

$$x(k+1) = x(k) + f(x(k), U(k)),$$

где

$$f(x(k), U(k)) = \Delta f_1(x(k), U(k)).$$

Теперь представим интеграл в виде суммы:

$$\sum_{k=0}^{N-1} Q_1(x(k), U(k))\Delta + \Psi_1(x(N\cdot\Delta)) = \sum_{k=0}^{N-1} Q(x(k), U(k)) + \Psi(x(N)).$$

В целом, после дискретизации задача описывается такой моделью:

$$F = \sum_{k=0}^{N-1} Q(x(k), U(k)) + \Psi(x(N)) \to \min,$$

$$x(k+1) = x(k) + f(x(k), U(k)), \qquad (I.4)$$

$$x(0) = x^{(0)}; \qquad (I.5)$$

$$U(k) \in \Omega(U). \qquad (I.6)$$

Для решения этой задачи применим так называемую обратную прогонку, т.е. прием, заключающийся в обратном движении по фазовой траектории от ее конца к началу, что соответствует переходу от $t = T$ к t_0. Начнем анализировать поведение объекта с момента времени $t = (N-1)\Delta$; этому моменту соответствует некоторое состояние $x(N-1)$ объекта. При этом нас не интересует, каким образом система пришла в это состояние под действием всех предыдущих управлений $U(0), U(1\Delta), \ldots, U((N-2)\Delta)$. Рассмотрим последний интервал, для которого $(N-1)\Delta \leq t \leq N\Delta$. В соответствии с принципом оптимальности будем оптимизировать поведение объекта на этом последнем интервале, т.е. будем решать задачу

$$F_{N-1} = Q(x(N-1), U(N-1)) + \Psi(x(N)) \to \min$$

при ограничениях (I.4) – (I.6). Из выражения (I.4) следует, что

$$x(N) = x(N-1) + f(x(N-1), U(N-1)).$$

Обозначим с помощью S_{N-1} наименьшее значение F_{N-1}, т.е.

$$S_{N-1} = \min_{U(N-1) \in \Omega(U)} F_{N-1}.$$

Здесь величина S_{N-1} при каждом конкретном значении $x(N-1)$ зависит только от одного параметра – от искомого управления $U(N-1)$. Мы понимаем, конечно, что к моменту $(N-1)\Delta$ значения $x(N-1)$ определяются всей совокупностью предшествующих управлений и, следовательно, эти состояния могут быть самыми различными среди некоторого множества допустимых состояний $X(N-1)$; специально еще раз отметим, что эти состояния определяются, таким образом, "предысторией" процесса управления при всех $0 \leq t \leq (N-1)\Delta$. Следовательно, для каждого $x(N-1) \in X(N-1)$ мы должны определить

$$S_{N-1}(x(N-1)) = \min_{U(N-1) \in \Omega(U)} \{Q(x(N-1), U(N-1)) + \Psi(x(N-1) + f(x(N-1), U(N-1)))\}.$$

В результате подобных вычислений мы получаем перечень значений S_{N-1} в функции от $x(N-1)$ и перечень соответствующих значений управляющего воздействия $\tilde{U}(N-1)$ в функции от тех же са-

мых состояний $x(N-1)$. Величины $\tilde{U}(N-1)$ называются условно-оптимальными управлениями; каждое из этих управлений $\tilde{U}(N-1)$ является оптимальным для перевода объекта в конечное состояние при условии, что к началу последнего интервала управления объект находится в состоянии $x(N-1)$. Каким-либо образом запоминаем все рассмотренные состояния $x(N-1)$ и соответствующие этим состояниям условно-оптимальные выигрыши S_{N-1} и условно-оптимальные управления $\tilde{U}(N-1)$.

Теперь перейдем к моменту времени $t=(N-2)\Delta$ начала последнего интервала и рассмотрим процесс управления совместно, т.е. как единое целое, на двух интервалах – предпоследнем и последнем. Для этих двух интервалов совместно общая эффективность описывается выражением

$$F_{N-2} = Q(x(N-2), U(N-2)) + \{Q(x(N-1), U(N-1)) + \Psi(x(N))\} \to \min,$$

в котором выигрыш на предпоследнем интервале зависит только от одной переменной $U(N-2)$, а выигрыш на последнем интервале – от одной переменной $U(N-1)$. Следовательно, мы должны записать, что

$$S_{N-2}(x(N-2)) = \min_{\substack{U(N-2)\in\Omega(U)\\ U(N-1)\in\Omega(U)}} F_{N-2} = \min_{U(N-2)\in\Omega(U)} \{Q(x(N-2), U(N-2)) + S_{N-1}(x(N-1))\}.$$

С учетом выражения (I.5) окончательная запись условно-оптимального выигрыша имеет вид

$$S_{N-2}(x(N-2)) = \min_{U(N-2)\in\Omega(U)} \{Q(x(N-2), U(N-2)) + S_{N-1}(x(N-2) + f(x(N-2), U(N-2)))\}.$$

Мы видим, что это выражение зависит уже только от одного аргумента $U(N-2)$ для каждого фиксированного состояния $x(N-2)\in X(N-2)$. Итак, мы получили условно-оптимальное управление (и условно-оптимальный выигрыш) на двух последних интервалах, т.е. минимизировали управление на всем отрезке от $(N-2)\Delta$ до $N\Delta$. Теперь каким-либо образом запоминаем все рассмотренные возможные состояния $x(N-2)$ перед началом предпоследнего шага и соответствующие значения $\tilde{U}(N-2)$ и S_{N-2}. Следовательно, мы для каждого

ле нахождения оптимального управления $\tilde{U}(0)$ необходимо осуществить так называемую прямую прогонку, т.е. просмотр решения в прямом направлении, от первого этапа к последнему, учитывая, что ранее мы запомнили наборы значений $x(k) \in \tilde{X}(k)$, $\tilde{U}(k)$, S_k для $k = 0, 1, \ldots, N-1$. В результате определим весь ряд оптимальных управлений и соответствующих состояний:

$$x(0) \to U^*(0) \to x^*(1) \to U^*(1) \to x^*(2) \to U^*(2) \to \ldots \to x^*(N-1) \to U^*(N-1).$$

Рассмотренный процесс решения принципиально переносится на исследование объекта любого порядка n с управлением порядка r. Нужно лишь заменить в приведенных выше формулах скаляры x, U и f на векторы X, U, f для каждого момента времени $t = (N-k)\Delta$:

$$X(N-k) = \begin{pmatrix} x_1(N-k) \\ x_2(N-k) \\ \ldots \\ x_n(N-k) \end{pmatrix}, \quad U(N-k) = \begin{pmatrix} U_1(N-k) \\ U_2(N-k) \\ \ldots \\ U_r(N-k) \end{pmatrix}, \quad f = \begin{pmatrix} f_1 \\ f_2 \\ \ldots \\ f_n \end{pmatrix}.$$

При этом рекуррентная формула остается прежней:

$$S_{N-k}(x(N-k)) = \min_{U(N-k) \in \Omega(U)} \{ Q(x(N-k), U(N-k)) + S_{N-k+1}(x(N-k) + f(x(N-k), U(N-k))) \}.$$

Это выражение называется функциональным уравнением Беллмана, или функциональным уравнением динамического программирования для дискретных систем, объектов, процессов. Фактически, такое уравнение задает алгоритм вычислений при решении задачи. В результате этих вычислений определяются параметры $U^*(j)$, $x^*(j)$ для $j = 0, 1, 2, \ldots, N-1$. Разумеется, в векторном случае на каждом этапе вычислений требуется находить минимум функции F_{N-k} от r переменных $U_1(N-k)$, $U_2(N-k)$, ..., $U_r(N-k)$. В свою очередь, оптимальные величины S_{N-k}, $U^*(N-k)$ являются функциями вектора $x(N-k)$, т.е. функциями n переменных $x_1(N-k)$, $x_2(N-k)$, ..., $x_n(N-k)$.

Обычно удобно представлять весь процесс вычислений в виде таблиц или графиков, но иногда удается провести все вычисления в аналитической форме.

Мы видим, что динамическое программирование фактически позволяет заменить оптимизацию сложной исходной задачи последовательностью оптимизаций существенно более простых задач. Разумеется, такой процесс упрощения исходной задачи достается не "даром", а вызывает увеличение числа шагов (стадий) решения.

55. Метод динамического программирования для непрерывных систем.

В этом разделе мы рассмотрим формализованное представление принципа оптимальности динамического программирования для непрерывных систем. Допустим, что движение объекта описывается векторным уравнением

$$\frac{dx}{dt} = f(x, U, t).$$

В начальный момент времени t_0 вектор $x(t)$ равен

$$x\big|_{t=t_0} = x^{(0)} \qquad (I.7)$$

Критерий оптимальности имеет вид

$$F = \int_{t_0}^{T} Q(x, U, t)\, dt \to \min,$$

где $U(t)$ – вектор r-й размерности, т.е. вектор управляющих воздействий. Для упрощения рассуждений предположим $T = const$. Пусть найдена оптимальная траектория $x^*(t)$, $t_0 \le t \le T$ перехода объекта из начального состояния $x^{(0)}$ в конечное состояние $x^{(T)}$, и пусть найдено соответствующее этому переходу оптимальное управление $U^*(t)$, как изображено на рис. I.2.

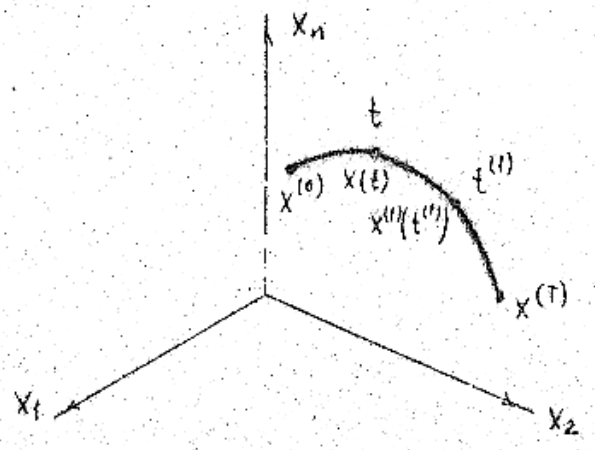


Рис. I.2

Рассмотрим некоторый момент времени $t > t_0$ и соответствующую точку на оптимальной траектории $x(t)$. Согласно принципу оптимальности, участок траектории от точки $x(t)$ до точки $x^{(T)}$ сам, в свою очередь, является оптимальной траекторией, и этому участку отвечает минимум части функционала, описывающей этот участок при изменении времени от t до T. Обозначим значение этого минимума $S(x(t), t)$. Возьмем малый промежуток времени Δt и момент времени $t^{(1)} = t + \Delta t$; этому моменту соответствуют точка $x^{(1)}(t^{(1)})$ на фазовой траектории и величина $S(x^{(1)}(t^{(1)}), t^{(1)}) = S(x(t+\Delta t), t+\Delta t)$, которая отвечает участку траектории от точки $x^{(1)}(t^{(1)})$ до точки $x^{(T)}$ и доставляет минимум части функционала, соответствующей этому участку.

Сформируем выражение, связывающее между собой параметры $S(x(t), t)$ и $S(x^{(1)}(t^{(1)}), t^{(1)})$. Это выражение аналогично функциональному уравнению Беллмана для дискретных систем, которое мы рассматривали выше и которое имело вид

$$S_{N-K}(x(N-K)) = \min_{U(N-K)\in\Omega(U)} \{Q(x(N-K), U(N-K)) + S_{N-K+1}(x(N-K) + f(x(N-K), U(N-K)))\}. \quad (I.8)$$

Для перевода этой записи в выражение для непрерывного случая заменим в нем $S_{N-K}(x(N-K))$ на $S(x,t)$, $S_{N-K+1}(x(N-K+1))$ на $S(x^{(1)}, t^{(1)})$, $Q(x(N-K))$, $U(N-K)$ на $Q(x(t), U(t), t)\Delta t$. Кроме того, для учета неточности замены дифференциального уравнения соотношением в конечных разностях необходимо ввести в получаемое выражение величину порядка малости выше Δt, т.е. величину $Q_1(\Delta t)$, для которой выполняется предельное соотношение

$$\lim_{\Delta t \to 0} \frac{Q_1(\Delta t)}{\Delta t} = 0.$$

В итоге вместо выражения (I.II) получим следующую запись:

$$S(x,t) = \min_{U(t)\in\Omega(U)} \{Q(x(t), U(t), t)\Delta t + S(x^{(1)}, t^{(1)})\} + Q_1(\Delta t). \quad (I.9)$$

Это же уравнение можно получить и другим путем, безотносительно к рассмотренному выше дискретному случаю. Для этого, согласно определению, мы должны записать, что

$$S(x,t) = \min_{U(t)\in\Omega(U)} \int_t^T Q(x,U,t)dt = \min_{U(t)\in\Omega(U)} \left\{ Q(x,U,t)\Delta t + \int_{t^{(1)}}^T Q(x,U,\tau)d\tau \right\} + Q_1(\Delta t).$$

Этот интеграл можно представить в виде следующей суммы слагаемых:

$$S(x,t) = \min_{U(t)\in\Omega(U)} \left\{ Q(x,U,t)\Delta t + \min_{U(\tau)\in\Omega(U)} \int_{t^{(1)}}^T Q(x,U,\tau)d\tau \right\} + Q_1(\Delta t),$$

где Δt – малая величина, а $Q_1(\Delta t)$ – величина порядка малости выше, чем Δt. Здесь первое слагаемое зависит от управления в момент t, и только второе слагаемое (интеграл) зависит еще и от управления $U(\tau)$ в интервале изменения t от $t^{(1)}$ до T. Тогда можно записать:

$$S(x,t) = \min_{U(t)\in\Omega(U)} \left\{ Q(x,U,t)\Delta t + S(x^{(1)}, t^{(1)}) \right\} + Q_1(\Delta t). \qquad (I.10)$$

Мы видим, что формулы (I.9) и (I.10) совпадают. Теперь получим соотношение, связывающее между собой параметры $S(x,t)$ и $S(x^{(1)}, t^{(1)})$. Для этого проведем разложение в ряд Тейлора функции $x^{(1)}$ в окрестности точки $x(t)$ и в результате получим запись, конечно, с учетом выражения (I.9):

$$x^{(1)} = x(t+\Delta t) = x(t) + \frac{dx}{dt}\Delta t + Q_2(\Delta t) = x(t) + f(x,U,t)\Delta t + Q_2(\Delta t), \qquad (I.11)$$

где величина $Q_2(\Delta t)$ высшего порядка малости по сравнению с малым Δt. Подставим выражение $x^{(1)}$ из (I.11) в соотношение $S(x^{(1)}, t^{(1)})$ и получим запись:

$$S(x^{(1)}, t^{(1)}) = S(x(t) + f(x,U,t)\Delta t + Q_2(\Delta t), t+\Delta t).$$

Теперь предположим, что функция $S(x,t)$ в действительности существует, непрерывна и имеет частные производные по всем переменным x_j ($j = 1, 2, \ldots, n$) и по t, т.е. существуют $\partial S/\partial x_j$ ($j = 1, 2, \ldots, n$) и $\partial S/\partial t$. При справедливости этого допущения являются справедливыми и последующие рассуждения. Если же такое допущение не выполняется, то последующие построения имеют лишь интуитивно верный характер. Действительно, в некоторых непрерывных системах такое допущение не выполняется, и в этих случаях применение динамического программирования нуждается в дополнительном обосновании. Итак, примем справедливость допущения и разложим функцию $S(x^{(1)}, t^{(1)})$

в ряд Тейлора в окрестности точки $(x(t), t)$:

$$S(x^{(1)}, t^{(1)}) = S(x,t) + \sum_{j=1}^{n} \frac{\partial S(x,t)}{\partial x_j} f_j(x, U, t)\Delta t + \frac{\partial S(x,t)}{\partial t}\Delta t + Q_3(\Delta t) =$$

$$= S(x,t) + (\operatorname{grad} S(x,t), f(x,U,t)\Delta t) + \frac{\partial S(x,t)}{\partial t}\Delta t + Q_3(\Delta t).$$

Подставим это выражение вместо $S(x^{(1)}, t^{(1)})$ в соотношение (I.I0) и перенесем в левую часть элементы $S(x,t)$ и $\partial S/\partial t$, поскольку они не зависят от управления $U(t)$. Разделив все выражение на Δt, получим

$$-\frac{\partial S(x,t)}{\partial t} = \min_{U(t)\in\Omega(U)} \{Q(x,U,t) + (\operatorname{grad} S(x,t), f(x,U,t))\} + \frac{Q_4(\Delta t)}{\Delta t}.$$

В результате предельного перехода при $\Delta t \to 0$ получим соотношение

$$-\frac{\partial S(x,t)}{\partial t} = \min_{U(t)\in\Omega(U)} \{Q(x,U,t) + (\operatorname{grad} S(x,t), f(x,U,t))\}. \qquad (I.I2)$$

Это выражение и является уравнением Беллмана для непрерывных систем. В последних выражениях для упрощения их записи использован градиент функции $S(x,t)$ – вектор с координатами

$$\left(\frac{\partial S(x,t)}{\partial x_1}, \frac{\partial S(x,t)}{\partial x_2}, \ldots \frac{\partial S(x,t)}{\partial x_n} \right)^T.$$

В качестве примера рассмотрим задачу, для которой $Q(x)$ не зависит явно от t, и содержащую две переменные состояния x_1, x_2 и одну переменную управления U. Пусть уравнения объекта следующие:

$$\frac{dx}{dt} = f_1(x,U) = Ux_1 + x_2, \quad \frac{dx_2}{dt} = f_2(x,U) = U^2.$$

Тогда функциональное уравнение (I.I2) принимает вид

$$-\frac{\partial S}{dt} = \min_U \left[Q(x_1, x_2) + \frac{\partial S}{\partial x_1}(Ux_1 + x_2) + \frac{\partial S}{\partial x_2} U^2 \right]. \qquad (I.I3)$$

Предположим, что $\partial S/\partial x_2 > 0$, и найдем минимум выражения в фигурной скобке, приравняв для этого нулю ее производную по U. Тогда оптимальное значение U^*, минимизирующее эту скобку,

$$U^* = -\frac{1}{2} x_1 \frac{\partial S}{\partial x_1} \cdot \frac{1}{\frac{\partial S}{\partial x_2}}.$$

Подставим это соотношение в выражение (I.I3) и получим уравнение в частных производных:

$$-\frac{\partial S}{\partial t} = Q(x_1, x_2) + \frac{\partial S}{\partial x_1} x_2 - x_1^2 \frac{\left(\frac{\partial S}{\partial x_1}\right)^2}{4 \frac{\partial S}{\partial x_2}}. \qquad (I.I4)$$

Пусть в рассматриваемой задаче критерием является выражение

$$F = \int_{t_0}^{T} Q(x)dt + \Psi_1(x(T)) \to \min.$$

Из этого соотношения можно получить граничные условия для решения уравнения (I.I4). Действительно, $S(x,T)$ является известной функцией, поскольку для момента времени $t_0 = T$ она равна известной функции $\Psi_1(x(T))$, так как при $t_0 = T$ интеграл равен нулю. Зная граничную функцию $S(x,T)$, можно каким-либо образом проинтегрировать уравнение (I.I4). Одним из обычных методов приближенного интегрирования является дискретизация задачи и решение полученных рекуррентных соотношений типа (I.8).

56. Решение задач распределения ресурсов методом динамического программирования.

Рассмотрим приложение динамического программирования для решения различных типичных задач распределения ресурсов.

С целью унификации описания введем систему обозначений. Пусть рассматриваются два процесса, протекающих в течение m этапов. Обозначим функции эффективности (дохода) каждого процесса на любом этапе функциями $f_i^{(1)}(z_i^{(1)})$, $f_i^{(2)}(z_i^{(2)})$, $i = 1, 2, \ldots, m$, а функции остатка (траты) вкладываемого в процесс ресурса на любом из этапов функциями $\psi_i^{(1)}(z_i^{(1)})$, $\psi_i^{(2)}(z_i^{(2)})$, $i = 1, 2, \ldots, m$. Величиной z_0 обозначим начальное значение ресурса. На каждом этапе текущий остаток ресурса распределяется между двумя процессами таким образом, чтобы получить наибольшее суммарное значение эффективности от обоих процессов за все m этапов.

В такой задаче, следовательно, функционал является обычной аддитивной функцией

$$f(z) = \sum_{i=1}^{m} f_i(z_i),$$

а управление представляет собой стратегию назначения текущего остатка ресурса для вложения в каждый из процессов: $z_i = z_i^{(1)} + z_i^{(2)}$.

Обозначим:

$f_i(z_i) = f_i^{(1)}(z_i^{(1)}) + f_i^{(2)}(z_i^{(2)})$ – значение функции эффективности на отдельно взятом i-м этапе;

$f_{i,m}(z)$ – значение функции эффективности на совместно рассматриваемых этапах с номерами i, $i+1$, $i+2$, ..., m;

$F_{i,m}(z)$ – условно-оптимальное значение функции эффективности на совместно рассматриваемых этапах с номерами i, $i+1$, $i+2$, ..., m;

$F^*_{i,m}(z)$ – оптимальное значение функции эффективности на совместно рассматриваемых этапах с номерами i, $i+1$, $i+2$, ..., m.

Согласно уравнению Беллмана для дискретных процессов, мы должны записать

$$f_{i,m}(z) = f_i(z_{i-1}, U_i) + F_{i+1,m}(z_i),$$

где z_{i-1} – значение ресурса, остающегося к концу $(i-1)$-го этапа (или к началу i-го этапа); U_i – управление на i-м шаге. Здесь подразумевается, что

$$z_i = \psi(z_{i-1}, U_i) = \psi_i^{(1)}(z_{i-1}^{(1)}, U_i) + \psi_i^{(2)}(z_{i-1}^{(2)}, U_i).$$

С учетом введенных обозначений получаем запись

$$f_{i,m}(z) = f_i(z_{i-1}, U_i) + F_{i+1,m}(\psi(z_{i-1}, U_i)).$$

Тогда система функциональных уравнений для решения этой задачи для процедуры обратной прогонки решения имеет следующий вид:

$$F_{m,m}(z) = \max_{U_m} \{f_m(z_{m-1}, U_m)\},$$

$$F_{i,m}(z) = \max_{U_i} \{f_i(z_{i-1}, U_i) + F_{i+1,m}(\psi(z_{i-1}, U_i))\}, \quad i = \overline{1, m-1}.$$

Теперь перейдем к рассмотрению некоторых типовых задач распределения ресурсов.

I. Рассмотрим задачу, в которой на каждом этапе распределение ресурсов между двумя процессами происходит полностью, т.е. без остатка. Будем считать также, что все функции эффективности и остатка являются однородными, т.е. такими, вид которых один и тот же независимо от номера этапа $i = \overline{1,m}$. Поэтому в обозначениях $f_i^{(1)}$, $f_i^{(2)}$, $\psi_i^{(1)}$, $\psi_i^{(2)}$ будем опускать индекс i. Тогда для этой задачи можно ввести следующие обозначения:

$$f_i(z) = f^{(1)}(z_{i-1}, U_i) + f^{(2)}(z_{i-1}, U_i),$$

$$\psi_i(z) = \psi^{(1)}(z_{i-1}, U_i) + \psi^{(2)}(z_{i-1}, U_i).$$

$$F_{m,m}(z) = \max_{U_m}\{f^{(1)}(z_{m-1}, U_m) + f^{(2)}(z_{m-1}, U_m)\},$$

$$F_{i,m}(z) = \max_{U_i}\{f^{(1)}(z_{i-1}, U_i) + f^{(2)}(z_{i-1}, U_i) + F_{i+1,m}(\psi^{(1)}(z_{i-1}, U_i) + \psi^{(2)}(z_{i-1}, U_i))\}, \quad i = \overline{1,m}$$

Как мы уже отмечали, в задачах данного типа распределение ресурсов означает назначение частей текущего ресурса процессам I и 2, что обозначается записью

$$z_i = z_i^{(1)} + z_i^{(2)}, \qquad z_i = z_i(x_i^{(1)}, x_i^{(2)}),$$

где $x_i^{(1)}$, $x_i^{(2)}$ – величины ресурса, вкладываемые в I-й и во 2-й процессы на i-м этапе. Фактически, разделение текущей величины ресурса z_i на $x_i^{(2)}$ и $x_i^{(2)}$ и является управлением U_i на i-м этапе. Тогда $z_{i-1} = x_i^{(1)} + x_i^{(2)}$, $f_i^{(1)} = f_i^{(1)}(x_i^{(1)})$, $f_i^{(2)} = f^{(2)}(x_i^{(2)})$. В свою очередь, величина ресурса, остающегося незатраченным к концу i-го этапа, рассчитывается следующим образом:

$$z_i = \psi(z_{i-1}, U_i) = \psi^{(1)}(x_i^{(1)}) + \psi^{(2)}(x_i^{(2)}).$$

Следовательно, основное функциональное уравнение имеет вид

$$F_{i,m}(z_{i-1}) = \max_{0 \le x_i^{(1)} \le z_{i-1}}\{f^{(1)}(x_i^{(1)}) + f^{(2)}(z_{i-1} - x_i^{(1)}) + F_{i+1,m}(\psi^{(2)}(x_i^{(1)}) + \psi^{(2)}(z_{i-1} - x_i^{(1)}))\},$$

где величина текущего ресурса z_{i-1} лежит в пределах $z_{i-1}^{(min)} \le z_{i-1} \le z_{i-1}^{(max)}$. Рассмотрим теперь подробно схему решения этой задачи, применяя прием обратной прогонки.

На последнем, m-м, этапе алгоритм необходимых вычислений описывается функциональным уравнением

$$F_{m,m}(z_{m-1}) = \max_{0 \le x_m^{(1)} \le z_{m-1}} \left\{ f^{(1)}(x_m^{(1)}) + f^{(2)}(z_{m-1} - x_m^{(1)}) \right\}, \quad z_{m-1}^{(min)} \le z_{m-1} \le z_{m-1}^{(max)}.$$

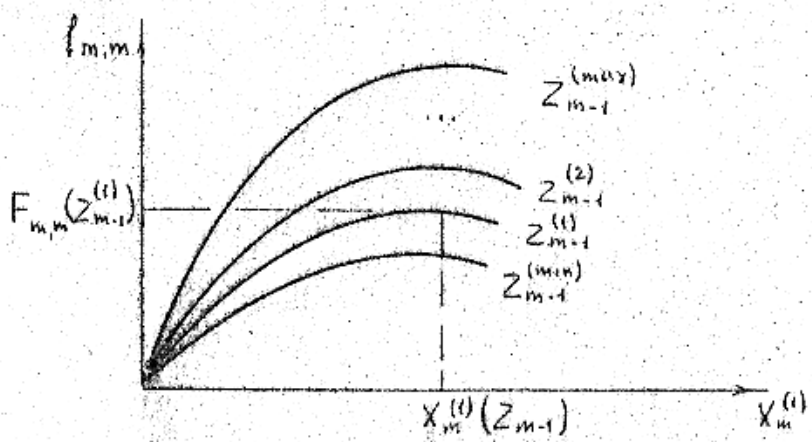
Мы понимаем, что состояние рассматриваемой системы, состоящей из двух процессов, в каждый момент времени описывается величинами $x_i^{(1)}$ и $x_i^{(2)}$, которые, в свою очередь, обобщенно обозначаются величиной $z_{i-1} = x^{(1)} + x^{(2)}$. Мы видим, что для решения функционального уравнения нам необходимо знать пределы возможных изменений текущей величины ресурса z_{i-1}, т.е. величин $z_{i-1}^{(min)}$ и $z_{i-1}^{(max)}$. Оказывается, эти величины очень просто определить. Для этого проведем следующие рассуждения. Пусть, например, для любого конкретного значения z справедливо соотношение $\psi^{(1)}(z) > \psi^{(2)}(z)$. Тогда величина $z_{m-1}^{(min)}$ определяется в соответствии с выражением

$$z_{m-1}^{(min)} = \psi^{(2)}\left(\psi^{(2)}\left(\psi^{(2)}\left(\dots(z_0)\right)\dots\right)\right),$$

представляющим собой сложную функцию, состоящую из $(m-1)$-й функции $\psi^{(2)}$. В свою очередь, величина $z_{m-1}^{(max)}$ рассчитывается по формуле

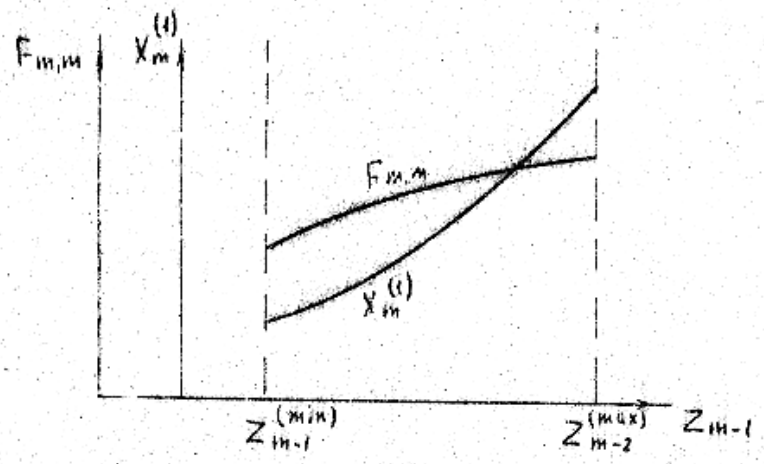
$$z_{m-1}^{(max)} = \psi^{(1)}\left(\psi^{(1)}\left(\psi^{(1)}\left(\dots(z_0)\right)\dots\right)\right).$$

Решение рассматриваемой задачи проведем графически. Разобьем весь интервал возможных значений z_{m-1} на ряд фиксированных значений: $z_{m-1}^{(min)}$, $z_{m-1}^{(1)}$, $z_{m-1}^{(2)}$, ..., z_{m-1}^{i}, $z_{m-1}^{(max)}$ и для каждого такого значения проведем вычисления в соответствии с функциональным уравнением. Результаты этих вычислений представим в виде семейства следующих графиков:



Для каждой кривой этого графика определим точки, в которых достигается наибольшая эффективность при данных z_{m-1} и $x_m^{(1)}$ и затем по-

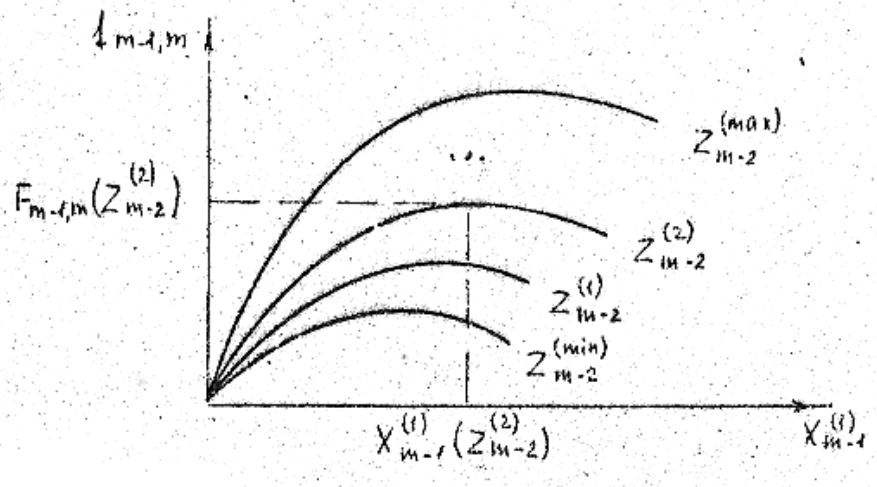
местим все эти точки на результирующий график, описывающий систему из двух процессов на m-м этапе:



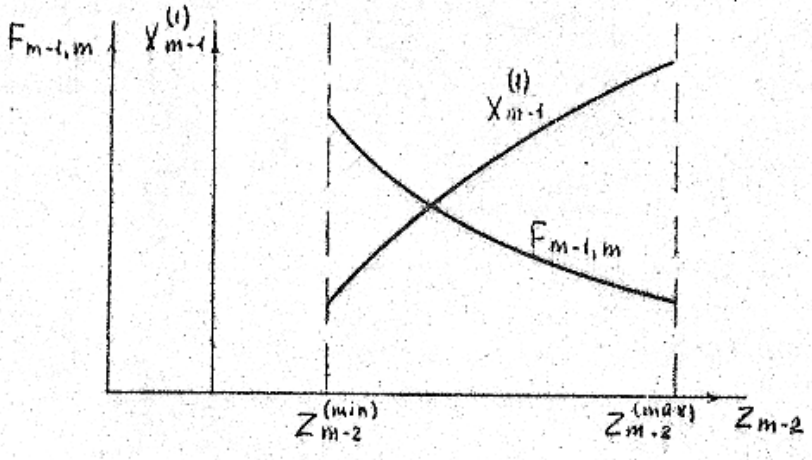
Для проведения дальнейших вычислений каким-либо образом запомним этот график. Теперь перейдем к следующему шагу вычислений, на котором будем рассматривать совместно предпоследний и последний этапы распределения ресурсов. Этому шагу вычислений соответствует функциональное уравнение

$$F_{m-1,m}(z_{m-2}) = \max_{0 \le x_{m-1}^{(i)} \le z_{m-2}} \left\{ f^{(1)}(x_{m-1}^{(i)}) + f^{(2)}(z_{m-2} - x_{m-1}^{(i)}) + F_{m,m}\left(\psi^{(1)}(x_{m-1}^{(i)}) + \psi^{(2)}(z_{m-2} - x_{m-1}^{(i)})\right)\right\},$$

для которого $z_{m-2}^{(min)} \le z_{m-2} \le z_{m-2}^{(max)}$. Разбиваем весь диапазон возможных значений z_{m-2} на фиксированные значения $z_{m-2}^{(min)}$, $z_{m-2}^{(1)}$, ..., $z_{m-2}^{(p)}$, $z_{m-2}^{(max)}$ и для каждого такого значения проведем вычисления согласно функциональному уравнению. Результаты этих вычислений поместим на следующем графике:



Отметим особо, что при вычислении каждого значения $f_{m-1,m}$ при каком-либо значении $X_{m-1}^{(n)}$ мы величины $f^{(1)}(X_{m-1}^{(1)})$ и $f^{(2)}(Z_{m-2}-X_{m-1}^{(1)})$ вычисляем, а значение $F_{m,m}(Z_{m-1})$ величины $Z_{m-1}=\psi^{(1)}(X_{m-1}^{(1)})+\psi^{(2)}(Z_{m-2}-X_{m-1}^{(1)})$ не вычисляем заново, а берем его из результирующего графика m-го этапа величины Z_{m-1}. Координаты получаемых таким образом экстремальных точек для всех кривых последнего графика мы для удобства сводим в единий результирующий график:



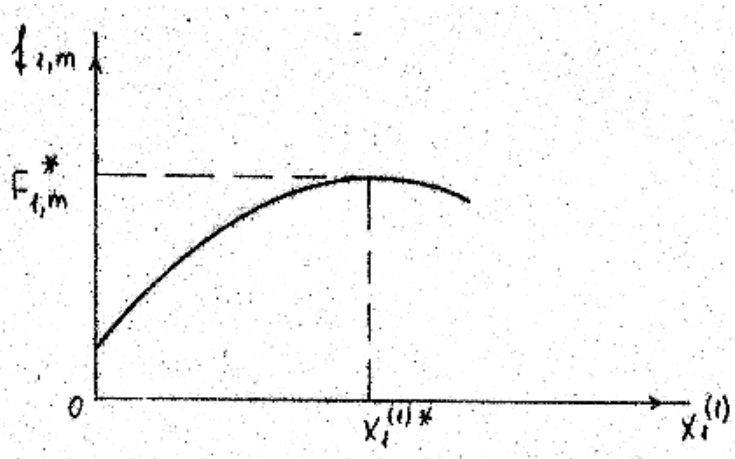
Запоминаем также и этот график для выполнения последующих вычислений.

Действуем аналогичным образом и приходим, наконец, к шагу вычислений, на котором рассматриваем распределение ресурсов совместно на всех этапах – от первого до последнего. Этому шагу вычислений соответствует функциональное уравнение

$$F_{1,m}(Z_0)=\max_{0\le X_1^{(1)}\le Z_0}\left\{f^{(1)}(X_1^{(1)})+f^{(2)}(Z_0-X_1^{(1)})+F_{2,m}(\psi^{(1)}(X_1^{(1)})+\psi^{(2)}(Z_0-X_1^{(1)}))\right\}.$$

Отметим специально, что величина текущего ресурса Z на этом этапе вычислений может принимать лишь одно значение, а именно, $Z=Z_0$. Таким образом, для этого шага вычислений общее число вычислительных операций уменьшается по сравнению с другими шагами вычислений. Возможный вид результирующего графика для данного этапа вычислений приведен на с.27.

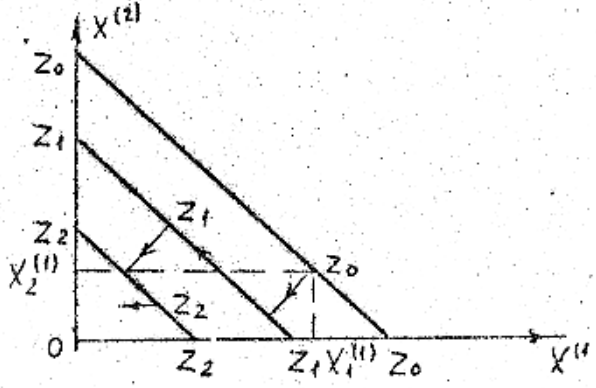
(т.е. следующей странице)



Точка $(F^*_{1,m}\ X_1^{(1)*})$ этого графика и является оптимальным решением задачи. Теперь мы осуществляем прямую прогонку решения, которая заключается в том, что мы из множества условно-оптимальных решений, которые ранее запомнили, отбираем уже истинное оптимальное решение. Такой процесс можно отобразить в виде следующей схемы действий:

$$Z_0 \to F^*_{1,m} \to X_1^{(1)*} \to Z_1^* \to X_2^{(1)*} \to Z_2^* \to X_3^{(1)*} \to \dots \to X_{m-1}^{(1)*} \to Z_{m-1}^* \to X_m^{(1)*}.$$

На этом решение рассматриваемой задачи считается законченным. На качественном уровне решение этой задачи можно отобразить с помощью диаграммы состояний системы, или фазовой диаграммы:



Отрезки прямых на диаграмме соответствуют текущим значениям ресурса перед началом каждого этапа, а участки фазовой траектории, принадлежащие этим отрезкам, отвечают моментам перераспределения текущего ресурса в начале очередного этапа.

2. Теперь опять рассмотрим задачу распределения ресурсов для управления двумя процессами на m этапах с начальной величиной ресурса z_0. Требуется достичь наибольшей суммарной эффективности за все m этапов. Но в этой задаче будем считать функции дохода

и остатка неоднородными, т.е. различными для каждого i -го этапа:

$$f_i^{(1)},\ f_i^{(2)},\ \psi_i^{(1)},\ \psi_i^{(2)},\quad i=\overline{1,m}.$$

Схема решения этой задачи ничем не отличается от схемы решения предыдущей задачи, только во всех выписанных ранее функциональных уравнениях на месте функций $f^{(1)}$, $f^{(2)}$ и $\psi^{(1)}$, $\psi^{(2)}$ должны стоять соответствующие шагу вычислений функции $f_i^{(1)}$, $f_i^{(2)}$ и $\psi_i^{(1)}$, $\psi_i^{(2)}$.

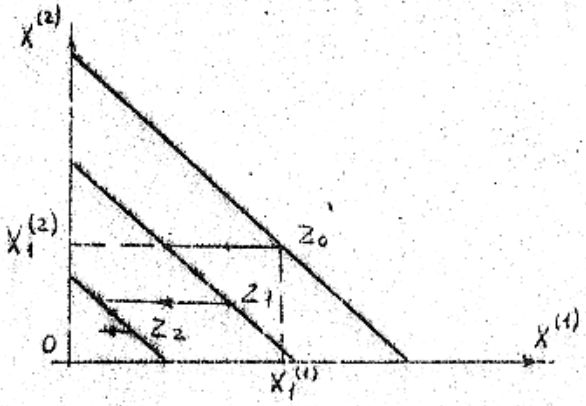
3. Теперь рассмотрим задачу распределения ресурсов на m этапах для системы, состоящей только из одного процесса, при начальной величине ресурса Z_0 с возможностью частичного или полного его резервирования для использования на последующих этапах. Целью управления по-прежнему является достижение наибольшей суммарной эффективности за все m этапов.

Приведем эту задачу к виду рассмотренных задач. Пусть реальный процесс описывается однородными функциями дохода и остатка $f^{(1)}(z)$ и $\psi^{(1)}(z)$. Для отражения возможности резервирования ресурса введем второй, фиктивный процесс, для которого функции дохода и остатка являются следующими однородными функциями:

$$f^{(2)}(z)=0\quad,\quad \psi^{(2)}(z)=z.$$

В такой интерпретации данная задача ничем не отличается от задачи, рассмотренной в п.I, и может быть решена по приведенной схеме вычислений.

Фазовая диаграмма, отображающая результаты решения рассматриваемой задачи, может иметь такой, например, вид:



На этой диаграмме горизонтальные участки фазовой траектории отображают трату (расход) ресурса, вкладываемого в первый процесс (реальный), и неизменность ресурса, вкладываемого во второй, фиктивный процесс на каждом этапе управления.

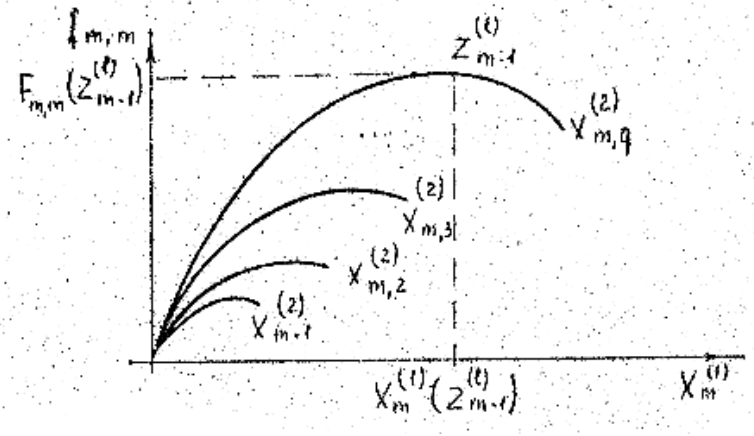
4. Рассмотрим конкретную задачу распределения ресурса

между тремя процессами на m этапах, причем будем подразумевать, что текущий ресурс распределяется между процессами полностью, т.е. без резервирования. Начальная величина ресурса обозначена значением z_0 ; функции дохода и остатка являются однородными: $f^{(1)}(z)$, $f^{(2)}(z)$, $f^{(3)}(z)$, $\psi^{(1)}(z)$, $\psi^{(2)}(z)$, $\psi^{(3)}(z)$. Требуется распределить ресурс таким образом, чтобы достичь наибольшей суммарной эффективности за все m этапов. Проводя рассуждения, аналогичные рассуждениям п.I, запишем функциональное уравнение для последнего этапа:

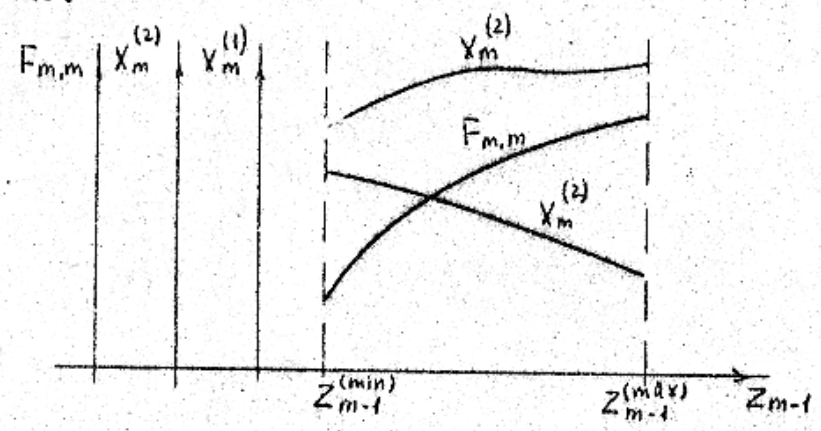
$$F_{m,m}(z_{m-1}) = \max_{0 \le x_m^{(1)} + x_m^{(2)} \le z_{m-1},} \left\{ f^{(1)}(x_m^{(1)}) + f^{(2)}(x_m^{(2)}) + f^{(3)}(z_{m-1} - x_m^{(1)} - x_m^{(2)}) \right\},$$

$$z_{m-1}^{(min)} \le z_{m-1} \le z_{m-1}^{(max)}.$$

Вычисления, задаваемые этим уравнением, интерпретируем с помощью графиков. Необходимо отметить, что для каждого возможного значения z_{m-1} конкретное значение $f_{m,m}$ зависит в данной задаче не от одной, а от двух переменных $x_m^{(1)}$ и $x_m^{(2)}$. Поэтому построения производим в следующем порядке. Разбиваем весь диапазон возможных значений ресурса z_{m-1} на ряд фиксированных значений $z_{m-1}^{(min)}$, $z_{m-1}^{(1)}$, $z_{m-1}^{(2)}$, ..., $z_{m-1}^{(t)}$, $z_{m-1}^{(max)}$. Допустим, что мы будем строить графики $f_{m,m}$ как функции от переменной $x_m^{(1)}$. Тогда для каждого конкретного значения z_{m-1} мы должны разбить весь диапазон возможных значений $x_m^{(2)}$, т.е. диапазон $0 \le x_m^{(2)} \le z_{m-1}$ на ряд фиксированных значений $x_{m,1}^{(2)}$, $x_{m,2}^{(2)}$, $x_{m,3}^{(2)}$, ..., $x_{m,q}^{(2)}$. Затем для каждого такого значения $x_{m,k}^{(2)}$ проводим соответствующие вычисления и строим одну зависимость $f_{m,m}$ при взятых значениях $z_{m-1}^{(t)}$ и $x_{m,k}^{(2)}$. Результаты всех таких вычислений могут иметь, например, вид следующих графиков:



Для каждого семейства кривых этих графиков отбираем наилучшее решение (только одну точку) и объединяем все эти точки в результирующем графике:



Запоминаем этот результирующий график и переходим к следующему шагу вычислений, на котором рассматриваем систему из трех процессов на последнем и предпоследнем этапах управления совместно. Соответствующие вычисления задаются функциональным уравнением

$$F_{m-1,m}(z_{m-2}) = \max_{0 \le x_{m-1}^{(1)} + x_{m-1}^{(2)} \le z_{m-2}} \left\{ f^{(1)}(x_{m-1}^{(1)}) + f^{(2)}(x_{m-1}^{(2)}) + f^{(3)}(z_{m-2} - x_{m-1}^{(1)} - x_{m-1}^{(2)}) + \right.$$

$$\left. + F_{m,m}(\psi^{(1)}(x_{m-1}^{(1)}) + \psi^{(2)}(x_{m-1}^{(2)}) + \psi^{(3)}(z_{m-2} - x_{m-1}^{(1)} - x_{m-1}^{(2)})) \right\}, \quad z_{m-2}^{(min)} \le z_{m-2} \le z_{m-2}^{(max)}.$$

Множество условно-оптимальных управлений для данного шага вычислений запоминаем в виде результирующего графика соответствующего вида. Аналогичным образом проводим вычисления и на остальных шагах решения задачи. Наконец приходим к шагу вычислений, на котором рассматриваем управление системой совместно на всех этапах от первого до последнего. Алгоритм вычислений для этого шага задается функциональным уравнением

$$F_{1,m}(z_0) = \max_{0 \le x_1^{(1)} + x_1^{(2)} \le z_0} \left\{ f^{(1)}(x_1^{(1)}) + f^{(2)}(x_1^{(2)}) + f^{(3)}(z_0 - x_1^{(1)} - x_1^{(2)}) + \right.$$

$$\left. + F_{2,m}(\psi^{(1)}(x_1^{(1)}) + \psi^{(2)}(x_1^{(2)}) + \psi^{(3)}(z_0 - x_1^{(1)} - x_1^{(2)})) \right\}.$$

Затем, как было описано выше, проводим прямую прогонку результатов решения и отбираем из условно-оптимальных управлений, которые запомнили ранее, истинное оптимальное управление. На этом решение задачи считается законченным.

Мы видим, что решение данной задачи является намного более трудоемким по сравнению с решением задачи из п.I, хотя число переменных состояния возросло всего только на одну переменную. Этот факт резкого возрастания трудоемкости решения даже при небольшом увеличении размерности задач Беллман образно назвал "проклятием размерности", свойственным методу динамического программирования, собственно как и другим численным методам.

5. Теперь рассмотрим задачу распределения ресурса между n процессами на m этапах. Начальная величина ресурса z_0. Но будем считать в этой задаче, что ресурс, вкладываемый в любой из процессов на любом этапе, тратится к концу этого этапа полностью, без остатка, т.е. $\psi^{(j)}(z) = 0 \cdot z = 0$, $j = \overline{1,n}$ (мы считаем все функции остатка однородными). Несколько преобразуем задачу, а именно, будем считать ее состоящей из $m\,n$ этапов, причем на каждом из таких этапов состояние системы характеризуется всего одной переменной состояния $x_i^{(j)}$; здесь индексы в скобках являются номерами реальных процессов, а индексы без скобок – номерами реальных этапов. Тогда совокупность всех переменных состояния от первого до последнего этапа из числа $m\,n$ этапов может быть представлена следующим рядом:

$$x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(n)}, x_2^{(1)}, x_2^{(2)}, \dots, x_2^{(n)}, \dots, x_m^{(1)}, x_m^{(2)}, \dots, x_m^{(n)}.$$

Пусть по-прежнему требуется так распределить ресурс, чтобы достичь наибольшей суммарной эффективности от всех процессов за все m реальных этапов. Обозначим функцию эффективности как $f_i^{(j)}(z)$, j-го процесса на i-м реальном этапе. В заменяющей задаче $m\,n$ этапов будем нумеровать от $\binom{(1)}{1}$ до $\binom{(n)}{m}$. Тогда для процедуры обратной прогонки функциональные уравнения будут иметь следующий вид:

$$F_{\binom{(n)}{m},\binom{(n)}{m}}(z) = \max_{\substack{0 \le x_m^{(n)} \le z \\ 0 \le z \le z_0}} \left\{ f_m^{(n)}\left(x_m^{(n)}\right) \right\},$$

$$F_{\binom{(n-1)}{m},\binom{(n)}{m}}(z) = \max_{\substack{0 \le x_m^{(n-1)} \le z \\ 0 \le z \le z_0}} \left\{ f_m^{(n-1)}\left(x_m^{(n-1)}\right) + F_{\binom{(n)}{m},\binom{(n)}{m}}\left(z_0 - x_m^{(n-1)}\right) \right\},$$

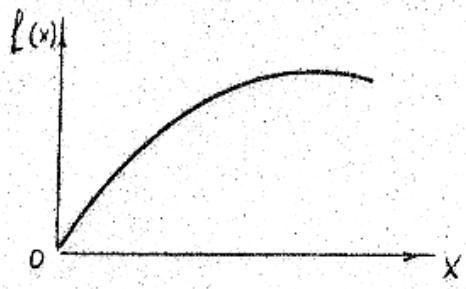
...

$$F_{\binom{(1)}{1},\binom{(n)}{m}}(z_0) = \max_{0 \le x_1^{(1)} \le z_0} \left\{ f_1^{(1)}(x_1^{(1)}) + F_{\binom{(2)}{1},\binom{(n)}{m}}(z_0 - x_1^{(1)}) \right\}.$$

Проведя затем прямую прогонку по множеству условно-оптимальных решений, получим оптимальное решение данной задачи.

Заметим, что задачи, в которых вкладываемый ресурс тратится к концу этапа без остатка, называются задачами с полной тратой ресурса.

6. Рассмотрим задачу распределения ресурса для одного процесса на m этапах. Начальная величина ресурса z_0. На каждом из этапов вложенный ресурс тратится полностью, т.е. $\psi_i(z) = 0$. Пусть функция дохода однородна и строго вогнута:



В этом случае для максимизации суммарной эффективности ресуро z_0 должен быть равномерно распределен по этапам. Покажем это, воспользовавшись методом математической индукции. Предварительно отметим, что производные $f'(z) > 0$ и $f''(z) < 0$.

Рассмотрим случай, когда задача включает только один этап распределения ресурса. При этом функциональное уравнение имеет вид

$$F_{1,1}(z_0) = \max_{0 \le x \le z_0} \{ f(x) \}.$$

Очевидно, что оптимальное управление, т.е. величина x_1^* равно z_0

Пусть теперь имеется задача, состоящая всего из двух этапов распределения ресурса. Для нее функциональное уравнение таково:

$$F_{1,2}(z_0) = \max_{0 \le x_1 \le z_0} \{ f(x_1) + F_{2,2}(z_0 - x_1) \}.$$

Найдем оптимальное решение такой задачи аналитическим способом. Для этого приравняем нулю первую производную функции $f_{1,2}(z_0)$ по переменной x_1:

$$\frac{d f_{1,2}(z_0)}{dx_1} = f'(x_1) - f'_{2,2}(z_0 - x_1) = 0.$$

Проанализируем знак второй производной этой функции по переменной x_1 :

$$\frac{d^2 f_{1,2}(z_0)}{dx_1^2} = f''(x_1) + f''_{2,2}(z_0 - x_1) < 0.$$

Следовательно, согласно достаточному условию оптимальности одномерных функций, существует максимальное значение функции в стационарной точке. Саму стационарную точку найдем из того условия, что для строго монотонной функции (а функция дохода является именно такой функцией) первые производные функции при различных аргументах равны, если равны сами аргументы. Тогда из соотношения $df_{1,2}(z_0)/dx_1 = 0$ получаем $x_1 = z_0 - x_1$. Следовательно, $x_1^* = \frac{z_0}{2}$, $F^*_{1,2}(z_0) = 2 f(\frac{z_0}{2})$.

Теперь допустим, что задача состоит в распределении ресурса на $(m-1)$ этапах и наше предположение о равномерном распределении ресурса выполняется. При этом

$$F^*_{1(m-1)}(z_0) = (m-1) f\left(\frac{z_0}{m-1}\right).$$

Перейдем к следующему шагу рассуждений и допустим, что теперь решается задача на m этапах. Для этой задачи функциональное уравнение для всей совокупности этапов имеет вид

$$F_{1,m}(z_0) = \max_{0 \le x_1 \le z_0} \{ f(x_1) + F_{2,m}(z_0 - x_1) \}.$$

Учитывая предыдущий шаг рассуждений, последнее соотношение можно записать в виде

$$F_{1,m}(z_0) = \max_{0 \le x_1 \le z_0} \{ f(x_1) + F^*_{1,(m-1)}(z_0 - x_1) \} =$$

$$= \max_{0 \le x_1 \le z_0} \left\{ f(x_1) + (m-1) f\left(\frac{z_0 - x_1}{m-1}\right) \right\}.$$

Для нахождения возможной стационарной точки и определения характера функции в этой точке определим первую и вторую производные функции по переменной x_1 :

$$f'_{1,m}(z_0) = f'(x_1) - f'\left(\frac{z_0 - x_1}{m-1}\right),$$

$$f''_{1,m}(z_0) = f''(x_1) + \frac{1}{m+1} f''\left(\frac{z_0 - x_1}{m-1}\right) < 0.$$

Судя по знаку второй производной, максимум имеется. Тогда приравняем нулю первую производную и определим величину x_1^* :

$$x_1 = \frac{z_0 - x_1}{m-1}, \text{ т.е. } m x_1 = z_0, \quad x_1^* = \frac{z_0}{m}.$$

Следовательно, $F^*_{1,m}(z_0) = m f\left(\frac{z_0}{m}\right)$. Поскольку свойство равномерного распределения ресурса по этапам выполняется для общего этапа наших рассуждений, постольку это свойство является справедливым. Значит, в задаче с полной тратой ресурса в случае строго вогнутой функции эффективности ресурс следует распределять равномерно по этапам.

7. Рассмотрим теперь задачу распределения ресурса в такой же формулировке, как в п.6, но со строго выпуклой однородной функцией дохода $f(z)$. Общее функциональное уравнение для этой задачи

$$F_{i,m}(z) = \max_{0 \le x_i \le z} \{f(x_i) + F_{i+1,m}(z - x_i)\}$$

включает в себя слагаемые, которые являются выпуклыми функциями от своих аргументов. Но мы знаем, что сумма выпуклых функций является, в свою очередь, выпуклой функцией. Интуитивно ясно, что она достигает своего максимума в одной или нескольких крайних точках выпуклого замкнутого множества, на котором задана эта выпуклая функция; данное свойство в теории математического программирования доказывается строгим образом. Таким выпуклым множеством в нашем случае является отрезок возможных значений переменной $x_i \in [0, z_0]$. Крайними точками этого множества являются точки $x_i = 0$ и $x_i = z_0$. Вычисляя значения функции в этих крайних точках, убеждаемся, что в такой задаче для достижения наибольшей суммарной эффективности ресурс необходимо вложить полностью на каком-нибудь из этапов.

8. Если в какой-либо задаче распределения ресурса необходимо добиться максимума общей функции эффективности для мультипликативного вида этой функции

$$f = \prod_{i=1}^{m} f_i(z_{i-1}, U_i),$$

то в такой задаче общее функциональное уравнение имеет следующую запись:

$$F_{i,m}(z_{i-1}) = \max_{U_i} \{f(z_{i-1}, U_i)\, F_{i+1,m}(z_i)\}.$$

В остальном схемы рассуждений при решении тех или иных задач распределения ресурса остаются аналогичными изложенным ранее схемам.

57. Решение задачи о коммивояжере методом динамического программирования.

Эта задача имеет следующую постановку. Имеется $n + 1$ городов; задана матрица расстояний между этими городами. Выезжая из исходного города, коммивояжер должен побывать во всех остальных городах ровно по одному разу и вернуться в исходный город. Требуется определить порядок, в котором коммивояжер должен объезжать города, чтобы общее пройденное расстояние было минимальным. Формализованная постановка этой задачи подразумевает задание матрицы расстояний. Такая матрица для задачи с пятью городами имеет следующий вид:

	X_0	X_1	X_2	X_3	X_4
X_0	∞	C_{01}	C_{02}	C_{03}	C_{04}
X_1	C_{10}	∞	C_{12}	C_{13}	C_{14}
X_2	C_{20}	C_{21}	∞	C_{23}	C_{24}
X_3	C_{30}	C_{31}	C_{32}	∞	C_{34}
X_4	C_{40}	C_{41}	C_{42}	C_{43}	∞

Несколько переформулируем задачу для применения метода динамического программирования. Пусть задано $n+1$ городов: $0, 1, 2, \ldots, n$. Задана также матрица расстояний $\{C_{ij}\}_{n,n}$; $i, j = \overline{0,n}$, $j \neq i$; $C_{ii} = \infty$, $i = \overline{0,n}$. Требуется найти перестановку $\bar{i} = (i_1, i_2, \ldots, i_n)$ чисел $1, 2, \ldots, n$, минимизирующую суммарный пройденный путь

$$f(x) = C_{0i_1} + C_{i_1 i_2} + C_{i_2 i_3} + \ldots + C_{i_{n-1} i_n} + C_{i_n 0} \to \min.$$

Пусть в качестве начального зафиксирован город 0. Обозначим $\{i_1, i_2, \ldots, i_{k-1}\}$ некоторые различные города, отличающиеся от начального города; здесь $i_s \neq i_t$ при $s \neq t$ и $i_s \neq 0$ для всех $s = 1, 2, \ldots, k-1$. Обозначим с помощью i_k город, отличный от городов $i_1, i_2, \ldots, i_{k-1}$. Заметим, что город i_k может совпадать с начальным городом ($i_k = 0$) только при $k = n+1$.

Обозначим с помощью $F_{k-1}(0; \{i_1, i_2, \ldots, i_{k-1}\}; i_k)$ длину кратчайшего пути, соединяющего город 0 с городом i_k и проходящего в каком-то порядке через города $i_1, i_2, \ldots, i_{k-1}$. Перестановку $(i'_1, i'_2, \ldots, i'_{k-1})$ городов $(i_1, i_2, \ldots, i_{k-1})$, на которой реализуется определяемый выражением F_{k-1} кратчайший путь, обозначим записью

$$J_{k-1}(0, \{i_1, i_2, \ldots, i_{k-1}\}; i_k) = (i'_1, i'_2, \ldots, i'_{k-1}).$$

Из выражений F_{k-1} и J_{k-1} непосредственно следует основная теорема алгоритма: пусть $(i'_1, i'_2, \ldots, i'_{k-1}) = J_{k-1}(0; \{i_1, i_2, \ldots, i_{k-1}\}; i_k)$ и $1 \leq t \leq k-1$; тогда $(i'_1, i'_2, \ldots i'_t) = J_t(0; \{i_1, i_2, \ldots, i_t\}; i_{t+1})$. Эта теорема может быть сформулирована и по-другому, исходя из общего принципа оптимальности динамического программирования, а именно: любой начальный участок оптимального маршрута сам в свою очередь является оптимальным маршрутом.

Фактически, эта теорема задает последовательность операций при решении задачи о коммивояжере методом динамического программирования. Будем задавать этот алгоритм с помощью функциональных уравнений динамического программирования.

0-й шаг. Вычисляем функцию

$$F_0(0;i) = C_{0i} \quad , \quad i = \overline{1,n}.$$

Общее число P_0 вычисляемых значений функции $F_0(0;i)$: $P_0 = n$. Для вычисления каждого одного значения функции требуется выполнить одно действие – выбор из матрицы расстояний. В результате выполнения этого шага вычислений для дальнейших операций нужно запомнить n значений функции $F_0(0;i)$: $q_0 = n$.

1-й шаг. Вычисляем функцию $F_1(0;\{i\};j)$ согласно функциональному уравнению

$$F_1(0;\{i\};j) = F_0(0;i) + C_{ij} \quad , \quad i,j = \overline{1,n} \quad , \quad i \neq j.$$

Для этого шага число вычисляемых значений функции равно

$$P_1 = n(n-1) = n C^1_{n-1}.$$

Для вычисления одного значения функции требуется выполнить три действия: два выбора чисел из матрицы расстояний и таблицы значений и одно сложение. После выполнения шага предыдущую таблицу значений $F_0(0;i)$ можно из памяти исключить, но только после окончательного завершения шага. Следовательно, необходимый объем памяти (до вычеркивания предыдущей таблицы)

$$q_1 = n + n(n-1).$$

Заметим, что в единице объема памяти мы фиксируем не только одно значение функции F_k , но и соответствующую этому значению перестановку Π_k .

k-й шаг. ($2 \leq k \leq n-1$). Вычисляем функцию $F_k(0;\{i_1, i_2, \ldots i_k\};i_{k+1})$ для всех возможных значений $i_1, i_2, \ldots, i_{k+1}$, причем $i_\tau \neq i_s$ при $\tau \neq s$, $i_\tau = 1, 2, \ldots, n$, $\tau = 1, 2, \ldots, k+1$. Вычисления проводим согласно функциональному уравнению

$$F_k(0;\{i_1,i_2,\ldots,i_k\};i_{k+1}) = \min\{F_{k-1}(0;\{i_2,i_3,\ldots,i_k\};i_1)+C_{i_1 i_{k+1}};$$
$$F_{k-1}(0;\{i_1,i_3,\ldots,i_k\};i_2)+C_{i_2 i_{k+1}};$$
$$\ldots$$
$$F_{k-1}(0;\{i_1,i_2,\ldots,i_{k-1}\};i_k)+C_{i_k i_{k+1}},$$

Для каждого значения функции $F_k(0;\{i_1, i_2, \dots, i_k\}; i_{k+1})$ запоминаем оптимальную перестановку $\Pi_k(0;\{i_1, i_2, \dots, i_k\}; i_{k+1})$. По окончании вычислений вычеркиваем из памяти таблицу значений

$F_{k-1}(0;\{i_1, i_2, \dots, i_{k-1}\}; i_k)$, при $k \geq 3$ также вычеркиваем и пути $\Pi_{k-1}(0;\{i_1, i_2, \dots, i_{k-1}\}; i_k)$

Для этого шага число вычисляемых значений функции F_k

$$p_k = n\, C_{n-1}^{k}.$$

На одно значение функции требуется выполнить $4k - 1$ действий: $2k$ выборов из таблиц, k сложений, $k - 1$ сравнений. Необходимый объем памяти (до вычеркивания из памяти значений F_{k-1} и Π_{k-1})

$$q_k = n\, C_{n-1}^{k-1} + n\, C_{n-1}^{k}.$$

n -й шаг. Проводим вычисления в соответствии с функциональным уравнением

$$F_n(0;\{1,2,\dots,n\};0) = \min \{F_{n-1}(0;\{2,3,\dots,n\};1) + C_{1,0};$$
$$F_{n-1}(0;\{1,3,\dots,n\};2) + C_{2,0};$$
$$\dots$$
$$F_{n-1}(0;\{1,2,\dots,n-1\};n) + C_{n,0}.$$

Перестановка $\Pi_n(0;\{1,2,\dots,n\};0) = (i_1', i_2', \dots i_n')$, на которой достигается минимум, является оптимальным решением исходной задачи. Для этого шага $p_n = 1$. Для одного значения функции требуется выполнить $4n - 1$ действий: $2n$ выборов из таблиц, n сложений, $n - 1$ сравнений. Необходимый объем памяти

$$q_n = n\, C_{n-1}^{n-1} + 1 = n+1.$$

На этом действия по алгоритму заканчиваются.

Оценим суммарный объем необходимых вычислений. Известно, что прямой перебор возможных вариантов решения задачи требует рассмотрения $n!$ вариантов и использования всего двух единиц памяти на любом этапе вычислений. Приближенно можно записать

$$n! \approx n^n e^{-n} \sqrt{2\pi n}$$, где $\pi = 3{,}14159$.

Работа же в соответствии с методом динамического программирования

требует выполнения общего числа операций

$$p = (6n-1) - n \cdot 2^{n-1} + 4n(n-1) \cdot 2^{n-2}$$

что существенно лучше в сравнении с методом прямого перебора. Максимально требуемый объем памяти для метода динамического программирования приближенно оценивается величиной

$$q \approx 2^n \sqrt{\frac{2n}{\pi}}, \qquad \text{где } \pi = 3{,}14159,$$

что по сравнению с прямым перебором существенно хуже.

Следовательно, метод динамического программирования позволяет достичь при решении задачи о коммивояжере существенного сокращения объема вычислений за счет заметного увеличения требуемого объема памяти.

Шпаргалка по 49-57 вопросам (кроме вопроса 52 «Жордановы исключения»)

Очень советую вместо шпор просто воспользоваться методами «Методы и алгоритмы решения линейных оптимизационных задач» и «Методы и алгоритмы решения нелинейных оптимизационных задач». К вопросам есть примеры, чтобы сократить писанину можно их отсечь.

Методичка «Методы и алгоритмы решения линейных оптимизационных задач»:

Вопрос 49:
Основные понятия о графах и сетях. «Методы и алгоритмы решения линейных оптимизационных задач» страницы **48-50**.
Метод решения задачи о кратчайшем пути. «Методы и алгоритмы решения линейных оптимизационных задач» страницы **55-57**.

Вопрос 50:
Метод Форда-Фалкерсона для решения задачи о максимальном потоке сети. «Методы и алгоритмы решения линейных оптимизационных задач» страницы **51-55**.

Вопрос 51:
Линейная сетевая задача, метод потенциалов для ее решения. «Методы и алгоритмы решения линейных оптимизационных задач» страницы **57-62**.

Методичка «Методы и алгоритмы решения нелинейных оптимизационных задач»:

Вопрос 53:
Задачи оптимального управления. Принцип оптимальности динамического программирования. «Методы и алгоритмы решения нелинейных оптимизационных задач» страницы **4-11**.

Вопрос 54:
Метод динамического программирования для дискретных систем. «Методы и алгоритмы решения нелинейных оптимизационных задач» страницы **11-17**.

Вопрос 55:
Метод динамического программирования для непрерывных систем. «Методы и алгоритмы решения нелинейных оптимизационных задач» страницы **17-21**.

Вопрос 56:
Решение задач распределения ресурсов методом динамического программирования. «Методы и алгоритмы решения нелинейных оптимизационных задач» стр. **21-35** (!!!)

Вопрос 57:
Решение задачи о коммивояжере методом динамического программирования. «Методы и алгоритмы решения нелинейных оптимизационных задач» страницы **40-46**.