

1 билет

1. Методы одномерного поиска.

Сущ. мн-во методов поиска минимума или максимума ф-ции на отрезке. Наиболее известными из них являются методы дихотомии (деления отрезка пополам), ЗС и Фибоначчи. В кажд. из этих методов последовательно сокращается интервал, содержащий точку минимума.

Метод дихотомии

Предполагаем, что минимизируемая ф-ция $f(x)$ унимодальна на отрезке $[a_0, b_0]$ и необх. найти минимум данной ф-ции на заданном отрезке с некоторой точностью ε . Вычисляем две точки согласно следующим ф-лам:

$$x_1 = \frac{a_0 + b_0 - \delta}{2} \text{ и } x_2 = \frac{a_0 + b_0 + \delta}{2}, \text{ где } \delta < \varepsilon.$$

И в каждой из найденных точек вычисляем знач-я ф-ции: $f(x_1)$ и $f(x_2)$.

Далее сокращаем интервал неопределенности и получаем интервал $[a_1, b_1]$ следующим образом. Ес. $f(x_1) < f(x_2)$, то $a_1 = a_0$ и $b_1 = x_2$.

В противном случае, ес. $f(x_1) > f(x_2)$, то $a_1 = x_1$ и $b_1 = b_0$.

Далее по аналог.ф-лам выч-ем следующую пару точек x_1 и x_2 . С пом-ю найденных точек определяем новый интервал неопределенности.

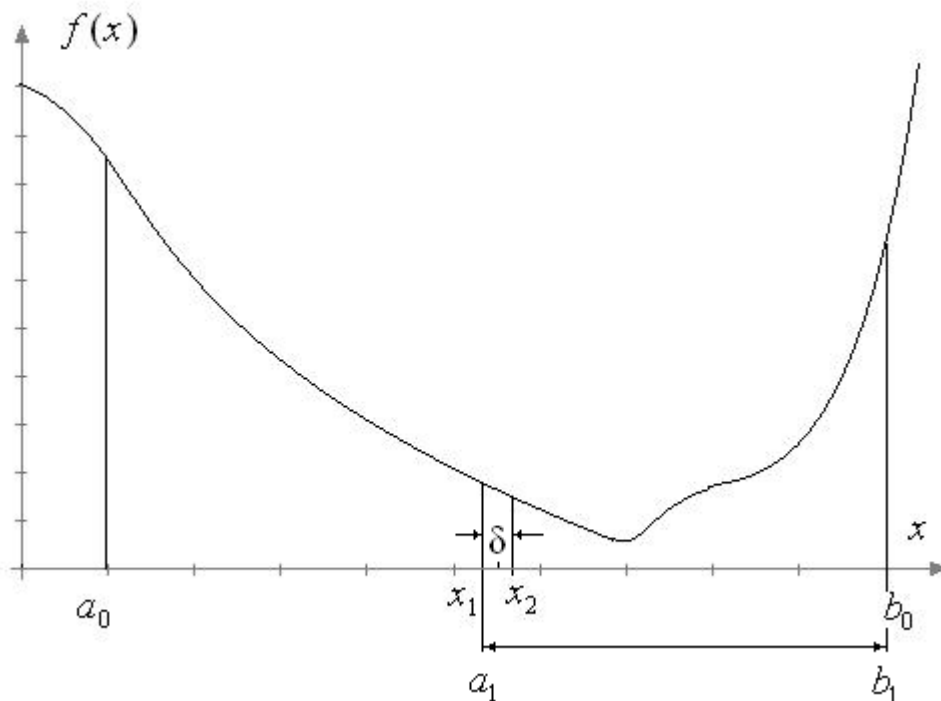


Рис. 1.

Поиск заканчивается, ес. длина интервала неопределенности $[a_k, b_k]$ на текущей итерации становится меньше заданной точности: $|b_k - a_k| < \varepsilon$.

В данном методе на каждой итерации минимизируемая ф-ция $f(x)$ вычисляется дважды, а интервал неопределенности сокращается практически в два раза (при малых $\delta < \varepsilon$).

Метод золотого сечения (ЗС)

Данный метод позволяет найти минимум заданной ф-ции на заданной области $[a_0, b_0]$, как правило, с меньшими вычисл-ми затратами, чем метод дихотомии. На первой итерации находим две точки по следующим

$$\text{ф-лам: } x_1 = a_0 + \frac{(3-\sqrt{5})}{2}(b_0 - a_0), \quad x_2 = b_0 + \frac{(\sqrt{5}-3)}{2}(b_0 - a_0)$$

и вычисляем значения ф-ции в них. Обратим внимание, что на первой итерации находим 2 точки и два вычисления $f(x)$.

Сокращаем интервал неопределенности.

- 1) Ес. $f(x_1) < f(x_2)$, то $a_1 = a_0$, $b_1 = x_2$, $x_2 = x_1$.
- 2) В противном случае, ес. $f(x_1) > f(x_2)$, то $a_1 = x_1$, $b_1 = b_0$, $x_1 = x_2$.

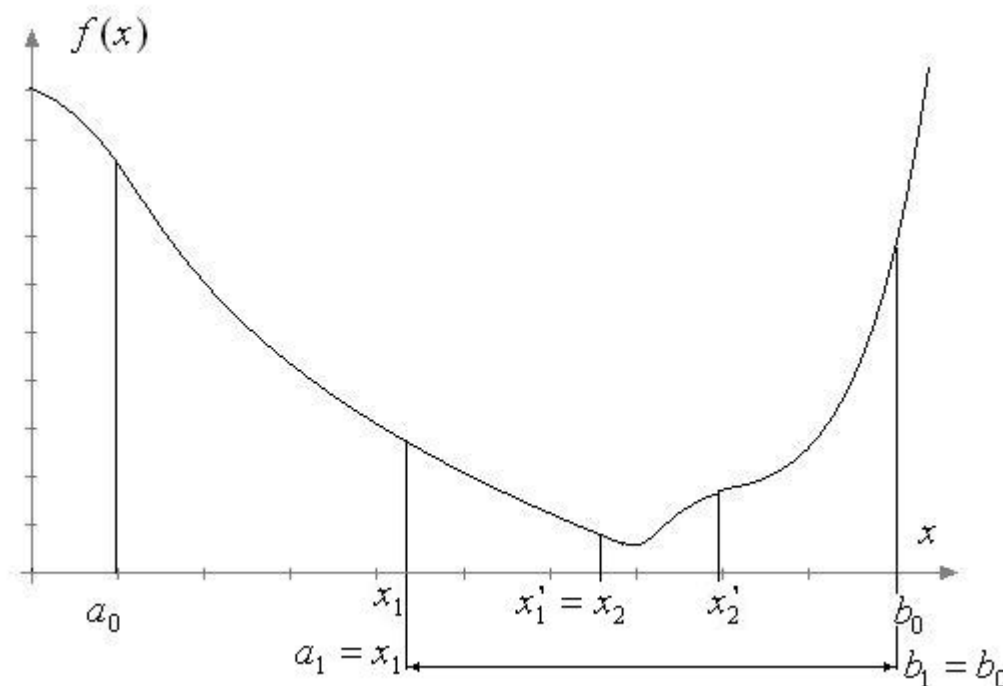


Рис. 2.

На последующих итерациях производим расчет только той точки и ее ф-ции, которую необх. обновить: в случае 1) вычисляем новое значение x_1 и $f(x_1)$; в случае 2) x_2 и $f(x_2)$.

Поиск прекращается при выполнении $|b_k - a_k| < \varepsilon$.

На i -й итерации интервал неопределенности сокращается до величины $0.6 \cdot (b_{i-1} - a_{i-1})$. Это меньше, чем в 2 раза, но зато всего один раз вычисляем значение $f(x)$ в новой точке.

Метод Фибоначчи

Числа Фибоначчи подчиняются соотношениям: $F_{n+2} = F_{n+1} + F_n$,

где $n = 1, 2, 3, \dots$ и $F_1 = F_2$. С помощью индукции можно показать, что n -е

число Фибоначчи вычисляется по ф-ле:
$$F_n = \frac{\left[\frac{1 + \sqrt{5}}{2} \right]^n - \left[\frac{1 - \sqrt{5}}{2} \right]^n}{\sqrt{5}},$$

где $n = 1, 2, 3, \dots$

Из данной формулы видно, что при больших значениях n

выполняется соотношение: $F_n \approx \frac{\left[\frac{1 + \sqrt{5}}{2} \right]^n}{\sqrt{5}}$, так, что числа Фибоначчи с увеличением n растут очень быстро.

Сам алгоритм метода Фибоначчи очень похож на алгоритм метода ЗС. На начальном интервале вычисляются точки по следующим ф-лам:

$$x_1 = a_0 + \frac{F_n}{F_{n+2}}(b_0 - a_0) \quad \text{и} \quad x_2 = a_0 + \frac{F_{n+1}}{F_{n+2}}(b_0 - a_0) = a_0 + b_0 - x_1.$$

Интервал неопределенности сокращ. так же, как в методе ЗС (рис.2). И на новой итерации выч-ся только 1 новая точка и значение ф-ции в ней.

На k -й итерации получаем точку минимума, кот. совпадает с одной из точек, которые вычисляются по ф-лам

$$x_1' = a_k + \frac{F_{n-k+1}}{F_{n-k+3}}(b_k - a_k) = a_k + \frac{F_{n-k+1}}{F_{n+2}}(b_0 - a_0),$$
$$x_2' = a_k + \frac{F_{n-k+2}}{F_{n-k+3}}(b_k - a_k) = a_k + \frac{F_{n-k+2}}{F_{n+2}}(b_0 - a_0)$$

и расположены на отр. $[a_k, b_k]$ симметрично относительно его середины.

При $k = n$ точки: $x_1' = a_n + \frac{F_1}{F_{n+2}}(b_0 - a_0)$ и $x_2' = a_n + \frac{F_2}{F_{n+2}}(b_0 - a_0)$

совпадают и делят отрезок $[a_n, b_n]$ пополам. $\Rightarrow, \frac{b_n - a_n}{2} = \frac{b_0 - a_0}{F_{n+2}} < \varepsilon$.

Отсюда можно выбрать n из условия $\frac{b_0 - a_0}{\varepsilon} < F_{n+2}$.

Т.о., это условие позволяет до начала работы алгоритма определить число итераций, необх.е для определения минимума с точностью ε при

начальной величине интервала $[a_0, b_0]$. С $\uparrow n$ из-за того, что F_n / F_{n+2} - бесконечная десятичная дробь, возможно "искажение" метода и потеря интервала с точкой минимума (вследствие погрешностей вычислений). Следует также отметить, что на практике метод ЗС по эффективности, скорости сх-ти и точности решения почти не уступает методу Фибоначчи. А алгоритмически реализация метода ЗС явл-ся более простой.

2.Первая теорема двойственности.Рассм. 2 двойственных задачи:

$$\left. \begin{array}{l} Q(\bar{x}) = \bar{p}^T \cdot \bar{x} \rightarrow \max \\ A \cdot \bar{x} \leq \bar{b} \\ \bar{x} \geq \bar{0} \end{array} \right\} (1) \quad \left. \begin{array}{l} W(\bar{u}) = \bar{b}^T \cdot \bar{u} \rightarrow \min \\ A^T \cdot \bar{u} \geq \bar{p} \\ \bar{u} \geq \bar{0} \end{array} \right\} (2)$$

Ес. одна из этих задач обладает оптимальным решением, то и двойственная к ней задача также имеет оптимальное решение. Причем экстремальные значения соответствующих линейных форм равны: $\max Q = \min W$.

Ес. же у одной из этих задач линейная форма не ограничена, то двойственная к ней задача противоречива.

Док-во: Пусть основная задача (1) имеет конечное решение и получена окончательная симплексная таблица:

		u_1 = $-y_1$	u_s = $-y_s$	v_{s+1} = $-x_{s+1}$	v_n = $-x_n$	W = 1
v_1	x_1 = .	$b_{1,1}$	$b_{1,s}$	$b_{1,s+1}$	$b_{1,n}$	$b_{1,n+1}$
....
v_s	x_s = .	$b_{s,1}$	$b_{s,s}$	$b_{s,s+1}$	$b_{s,n}$	$b_{s,n+1}$
u_{s+1}	y_{s+1} = .	$b_{s+1,1}$	$b_{s+1,s}$	$b_{s+1,s+1}$	$b_{s+1,n}$	$b_{s+1,n+1}$
....
u_m	y_m = .	$b_{m,1}$	$b_{m,s}$	$b_{m,s+1}$	$b_{m,n}$	$b_{m,n+1}$
1	Q = .	q_1	q_s	q_{s+1}	q_n	q_0

Т.к. данная таблица, по предположению, соответствует оптимальному решению задачи (1), то $b_{1,n+1} \geq 0, \dots, b_{m,n+1} \geq 0$ и $q_1 \geq 0, \dots, q_n \geq 0$. При этом $\max Q = q_0$ достигается при $y_1 = \dots = y_s = x_{s+1} = \dots = x_n = 0$.

Рассмотрим полученную таблицу двойственной задачи. Полагая значения переменных слева (небазисных) равными нулю:

$$v_1 = \dots = v_s = u_{s+1} = \dots = u_m = 0,$$

найдем $u_1 = q_1 \geq 0, \dots, u_s = q_s \geq 0, v_{s+1} = q_{s+1} \geq 0, \dots, v_n = q_n \geq 0$. \Rightarrow , получено опорное решение:

$$u_1 = q_1, \dots, u_s = q_s, u_{s+1} = 0, \dots, u_m = 0.$$

Из последнего столбца,

$$W = b_{1,n+1} \cdot v_1 + \dots + b_{s,n+1} \cdot v_s + b_{s+1,n+1} \cdot u_{s+1} + \dots + b_{m,n+1} \cdot u_m + q_0$$

в точке $v_1 = \dots = v_s = u_{s+1} = \dots = u_m = 0$

будет минимальным в силу того, что $b_{i,n+1} \geq 0 \quad \forall i, i = \overline{1, m}$. \Rightarrow , $\max Q = \min W$.

Пусть теперь линейная форма прямой задачи неограничена, т.е. для некоторой верхней переменной, например, y_s соответствующий коэффициент $q_s < 0$, а все коэффициенты этого столбца симплексной таблицы неположительны: $b_{1,s} \leq 0, b_{2,s} \leq 0, \dots, b_{m,s} \leq 0$. Тогда из таблицы для двойственной задачи:

$$u_s = b_{1,s} \cdot v_1 + \dots + b_{s,s} \cdot v_s + b_{s+1,s} \cdot u_{s+1} + \dots + b_{m,s} \cdot u_m + q_s \leq q_s < 0,$$

то есть система ограничений двойственной задачи противоречива. Т.к. из неотрицательности $v_1, \dots, v_s, u_{s+1}, \dots, u_m$ следует неположительность u_s (нельзя сделать ее положительной). То есть, система несовместна.

Теорема доказана.

2 билет

1.Прямые методы поиска. Алгоритм Гаусса. Алгоритм Хука и Дживса.

Прямые методы (методы нулевого порядка) не требуют знания целевой ф-ции в явном виде. Они не требуют регулярности и непрерывности целевой ф-ции и существования производных. Это явл-ся существенным достоинством при решении сложных технических и экономических задач.

При реализации прямых методов существенно сокращается этап подготовки решения задачи, т.к. нет необх.сти в определении первых и вторых производных. Прямые методы в основном носят эвристический

характер. К прямым методам относится целый ряд алгоритмов, которые отличаются по своей эффективности. Методы предназначены для решения безусловных задач оптимизации $\min_{\bar{x} \in E^n} f(\bar{x})$.

Алгоритм Гаусса

Это простейший алгоритм, заключающийся в том, что на каждом шаге (каждой итерации) минимизация осуществляется только по одной компоненте вектора переменных \bar{x} .

Пусть нам дано начальное приближение $\bar{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$. На первой итерации находим значение минимума ϕ -ции при изменяющейся первой координате и фиксированных остальных компонентах, т.е.

$$x_1^1 = \arg \min_{x_1} f(x_1, x_2^0, \dots, x_n^0).$$

В результате получаем новую точку $\bar{x}^1 = (x_1^1, x_2^0, \dots, x_n^0)$. Далее из точки \bar{x}^1 ищем минимум ϕ -ции, изменяя вторую координату и считая фиксированными все остальные координаты. В результате получаем значение

$$x_2^1 = \arg \min_{x_2} f(x_1^1, x_2, x_3^0, \dots, x_n^0)$$

и новую точку $\bar{x}^2 = (x_1^1, x_2^1, x_3^0, \dots, x_n^0)$. Продолжая процесс, после n шагов получаем точку $\bar{x}^n = (x_1^1, x_2^2, \dots, x_n^n)$, начиная с которой процесс возобновляется.

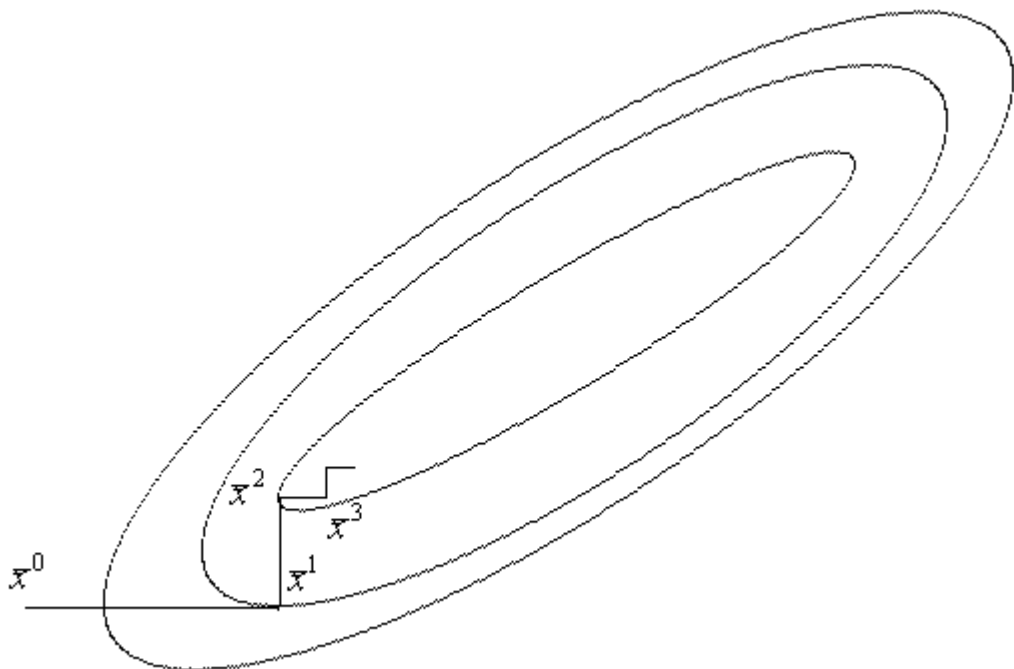


Рис. 1.

В качестве условий прекращения поиска можно использовать следующие два критерия:

- 1) $\|f(\bar{x}^{k+1}) - f(\bar{x}^k)\| \leq \varepsilon_0$.
- 2) $|x_i^{k+1} - x_i^k| \leq \varepsilon_i, \forall i$.

Метод очень прост, но не очень эффективен. Проблемы могут возникнуть, когда линии уровня сильно вытянуты и "эллипсоиды" ориентированы, например, вдоль прямых вида $x_1 = x_2$. В подобной ситуации поиск быстро застревает на дне такого оврага, а ес. начальное приближение оказывается на оси "эллипсоида", то процесс так и останется в этой точке.

Хорошие результаты получаются в тех случаях, когда целевая ф-ция представляет собой выпуклую сепарабельную функцию вида

$$f(\bar{x}) = \sum_{i=1}^n f_i(x_i) .$$

Алгоритм Хука и Дживса

В данном алгоритме предлагается логически простая стратегия поиска, в которой используются априорные сведения о топологии ф-ции и, в то же время отвергаются уже устаревшая информация об этой ф-ции. В интерпретации Вуда алгоритм включает два основных этапа:

- 1) исследующий поиск вокруг базисной точки \bar{x}^k ;
- 2) поиск по "образцу", т.е. в направлении, выбранном для минимизации.

Задается начальная точка поиска \bar{x}^0 и начальное приращение (шаг) $\Delta \bar{x}^0$, после чего начинается исследующий поиск.

Исследующий поиск:

Делаем пробный шаг по переменной x_1 , т.е. определяем точку $x_1^0 + \Delta x_1^0$ и вычисляем значение ф-ции для точки $\bar{x}' = (x_1^0 + \Delta x_1^0, x_2^0, \dots, x_n^0)$.

Ес. значение ф-ции в данной точке больше, чем значение ф-ции $f(\bar{x}^0)$, то делаем пробный шаг по этой же переменной, но в противоположном направлении. Ес. значение ф-ции в точке $\bar{x}'' = (x_1^0 - \Delta x_1^0, x_2^0, \dots, x_n^0)$ также больше, чем $f(\bar{x}^0)$, то оставляем точку x_1^0 без изменений. Иначе заменяем точку \bar{x}^0 на \bar{x}' или на \bar{x}'' в зависимости от того, где значение ф-ции меньше исходного. Из вновь полученной точки делаем пробные шаги по оставшимся координатам, используя тот же самый алгоритм.

Ес. в процессе исследующего поиска не удастся сделать ни одного удачного пробного шага, то $\Delta \bar{x}$ необх. скорректировать (уменьшить). После чего вновь переходим к исследующему поиску.

Ес. в процессе исследующего поиска сделан хоть один удачный пробный шаг, то переходим к поиску по образцу.

Поиск по образцу:

После исследующего поиска мы получаем точку \bar{x}^{01} . Направление $\bar{x}^{01} - \bar{x}^0$ определяет направление, в котором ф-ция уменьшается. Поэтому проводим минимизацию ф-ции в указанном направлении, решая задачу

$$\min_{\lambda} f(\bar{x}^0 + \lambda \cdot (\bar{x}^{01} - \bar{x}^0)).$$

В поиске по "образцу" величина шага по каждой переменной пропорциональна величине шага на этапе исследующего поиска. Ес. удастся сделать удачный шаг в поиске по "образцу", то в результате находим новое приближение $\bar{x}^1 = \bar{x}^0 + \lambda^0 \cdot (\bar{x}^{01} - \bar{x}^0)$, где

$$\lambda^0 = \arg \min_{\lambda} f(\bar{x}^0 + \lambda \cdot (\bar{x}^{01} - \bar{x}^0)).$$

Из точки \bar{x}^1 начинаем новый исследующий поиск и т.д.

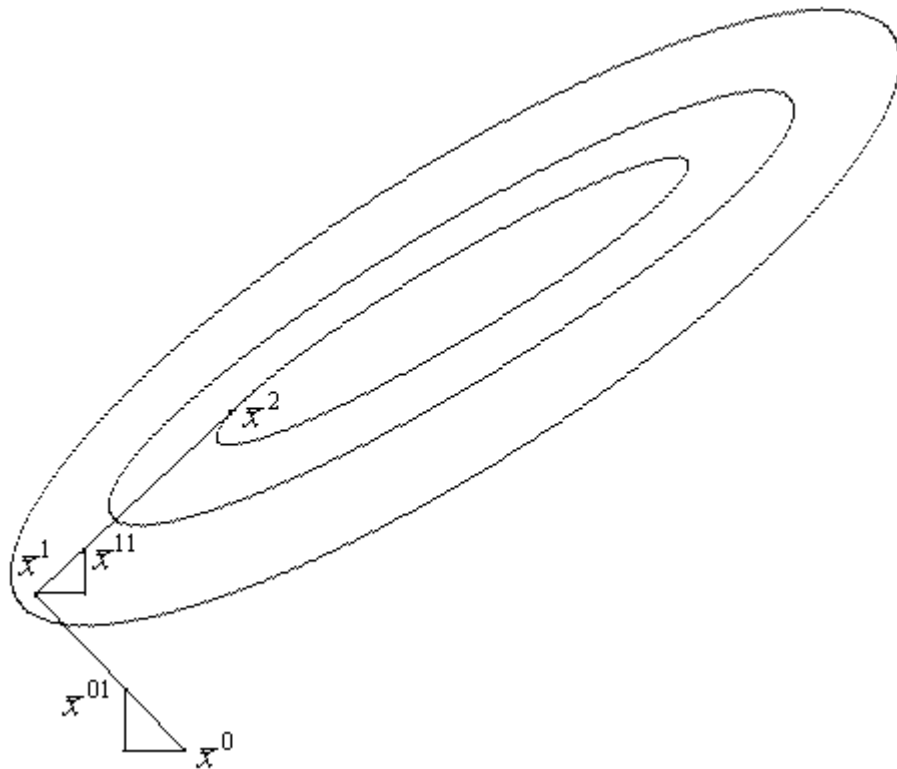


Рис. 2

Сущ-ют модификации алг-ма, в кот-х в процессе исследующего поиска ищется минимум по каждой переменной или в процессе поиска по образцу ищется не минимум ф-ции, а просто делается шаг в заданном найденном направлении с фикс. значением пар-ра λ .

2.Вторая теорема двойственности..

Ес. хотя бы одно оптимальное решение одной из двойственных задач обращает i -е ограничение этой задачи в строгое неравенство, то i -я компонента (т.е. x_i или u_i) каждого оптимального решения второй двойственной задачи равна нулю. Ес. же i -я компонента хотя бы одного оптимального решения одной из двойственных задач положительна, то каждое оптимальное решение другой двойственной задачи обращает i -е ограничение в строгое равенство.

Т.е. оптимальные решения \bar{x}^* и \bar{u}^* пары двойственных задач удовлетворяют условиям

$$x_j^* \cdot \left[\sum_{i=1}^m a_{ij} \cdot u_i^* - p_j \right] = 0, \quad j = \overline{1, n}, \quad (1)$$

$$u_i^* \cdot \left[\sum_{j=1}^n a_{ij} \cdot x_j^* - b_i \right] = 0, \quad i = \overline{1, m}. \quad (2)$$

Док-во: Пусть \bar{x}^* и \bar{u}^* – оптимальные решения пары двойственных задач.

Тогда для $Q(\bar{x}) = \sum_{j=1}^n p_j x_j \rightarrow \max$, $W(\bar{u}) = \sum_{i=1}^m b_i u_i \rightarrow \min$

они удовлетворяют следующим ограничениям:

$$\left. \begin{aligned} a_{i1} \cdot x_1^* + a_{i2} \cdot x_2^* + \dots + a_{in} \cdot x_n^* &\leq b_i, \quad i = \overline{1, m}, \\ x_j^* &\geq 0, \quad j = \overline{1, n}, \\ a_{1j} \cdot u_1^* + a_{2j} u_2^* + \dots + a_{mj} u_m^* &\geq p_j, \quad j = \overline{1, n}, \\ u_i^* &\geq 0, \quad i = \overline{1, m}. \end{aligned} \right\}. \quad (3)$$

Умножим (3), соответственно, на u_i^* и x_j^* , и просуммируем полученные

выражения: $\sum_{j=1}^n p_j \cdot x_j^* \leq \sum_{i=1}^m \sum_{j=1}^n a_{ij} \cdot u_i^* \cdot x_j^* \leq \sum_{i=1}^m b_i \cdot u_i^* .$ (4)

Из 1-ой теоремы двойственности $\Rightarrow \sum_{j=1}^n p_j \cdot x_j^* = \sum_{i=1}^m b_i \cdot u_i^* .$ (5)

И с учетом (4) получаем: $\sum_{j=1}^n p_j \cdot x_j^* = \sum_{j=1}^n \sum_{i=1}^m a_{ij} \cdot u_i^* \cdot x_j^* ,$

$$\sum_{i=1}^m b_i \cdot u_i^* = \sum_{i=1}^m \sum_{j=1}^n a_{ij} \cdot x_j^* \cdot u_i^* .$$

1-ое из этих выражений можем переписать: $\sum_{j=1}^n x_j^* \left(\sum_{i=1}^m a_{ij} \cdot u_i^* - p_j \right) = 0$,

и т.к. все x_j^* и выражения в скобках неотрицательны, то опуская \sum ,

получим: $x_j^* \cdot \left(\sum_{i=1}^m a_{ij} \cdot u_i^* - p_j \right) = 0, \quad j = \overline{1, n}$. Аналогично получим:

$u_i^* \cdot \left(\sum_{j=1}^n a_{ij} \cdot x_j^* - b_i \right) = 0, \quad i = \overline{1, m}$. Ч.т.д. Справедл. и обратная теорема.

3 билет

1.Алгоритм Розенброка.

Этот итерационный метод имеет некоторое сходство с алгоритмом Хука и Дживса. Метод Розенброка также называется методом вращающихся координат. Этот метод существенно эффективнее предыдущих методов, особенно при минимизации ф-ций овражного типа. Общая идея метода заключается в том, что выбирается система ортогональных направлений $\bar{S}_1^0, \bar{S}_2^0, \dots, \bar{S}_n^0$, в кажд.из которых последовательно ищется миним-е знач-е, после чего сис-ма направлений поворачивается так, чт.одна из осей совпала с направ-ем полного перемещ-я, а ост.были ортогональны м/у собой.

Пусть \bar{x}^0 - в-р нач-го приближения; $\bar{S}_1^0, \bar{S}_2^0, \dots, \bar{S}_n^0$ - сис-ма ортогональных направлений. На 1-ой итерации это м.б. ортонормированная сис-ма координат. Начиная с \bar{x}^0 , последов. осущ-ем минимизацию ф-ции $f(\bar{x})$ в соотв-щих направл-ях $\bar{S}_1^0, \bar{S}_2^0, \dots, \bar{S}_n^0$, находя последовательные приближ-я:

$$\bar{x}_1^0 = \bar{x}_0^0 + \lambda_1 \bar{S}_1^0, \quad \lambda_1 = \arg \min_{\lambda} f(\bar{x}_0^0 + \lambda \bar{S}_1^0),$$

....

$$\bar{x}_n^0 = \bar{x}_{n-1}^0 + \lambda_n \bar{S}_n^0, \quad \lambda_n = \arg \min_{\lambda} f(\bar{x}_{n-1}^0 + \lambda \bar{S}_n^0).$$

Следующая итерация начнется с точки $\bar{x}^1 = \bar{x}_n^0$. Ес. не изменить систему направлений, то будем иметь алгоритм Гаусса. => после заверш. очередного k -го этапа, вычисляем нов. направления поиска. Ортогональные направл-я поиска поворачиваются так, чт. они оказались вытянутыми вдоль "оврага" ("хребта") и, т.о., исключается взаимодействие переменных (x_i, x_j) . Направл-я поиска вытягиваются вдоль главн. осей квадратичной аппроксимации целевой ф-ции.

Рассм. Нек. k -ю итерацию алгоритма Розенбр. В результате минимизации по каждому из ортогональных направлений на данной итерации мы имеем сис-му параметров $\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k$, с пом-ю кот-х определим систему в-ров

$\bar{A}_1^k, \bar{A}_2^k, \dots, \bar{A}_n^k$, вычисляемых по ф-лам следующего вида:

$$\bar{A}_1^k = \lambda_1^k \bar{S}_1^k + \lambda_2^k \bar{S}_2^k + \dots + \lambda_n^k \bar{S}_n^k;$$

$$\bar{A}_2^k = \lambda_2^k \bar{S}_2^k + \dots + \lambda_n^k \bar{S}_n^k;$$

$$\dots;$$

$$\bar{A}_n^k = \lambda_n^k \bar{S}_n^k.$$

С помощью системы векторов $\bar{A}_1^k, \bar{A}_2^k, \dots, \bar{A}_n^k$ строим новую систему ортогональных направлений $\bar{S}_1^{k+1}, \bar{S}_2^{k+1}, \dots, \bar{S}_n^{k+1}$. Причем первый вектор направляют так, чт. он совпал с направлением общего перемещения на k -м шаге, а остальные получаются с помощью процедуры ортогонализации

Грама-Шмидта: $\bar{S}_1^{k+1} = \frac{\bar{A}_1^k}{\|\bar{A}_1^k\|};$

$$\bar{B}_2^k = \bar{A}_2^k - [(\bar{A}_2^k)^T \bar{S}_1^{k+1}] \bar{S}_1^{k+1};$$

$$\bar{S}_2^{k+1} = \frac{\bar{B}_2^k}{\|\bar{B}_2^k\|}; \quad (1)$$

$$\bar{B}_l^k = \bar{A}_l^k - \sum_{m=1}^{l-1} [(\bar{A}_l^k)^T \bar{S}_m^{k+1}] \bar{S}_m^{k+1}; \quad \bar{S}_l^{k+1} = \frac{\bar{B}_l^k}{\|\bar{B}_l^k\|}; \quad l = 2, \dots, n$$

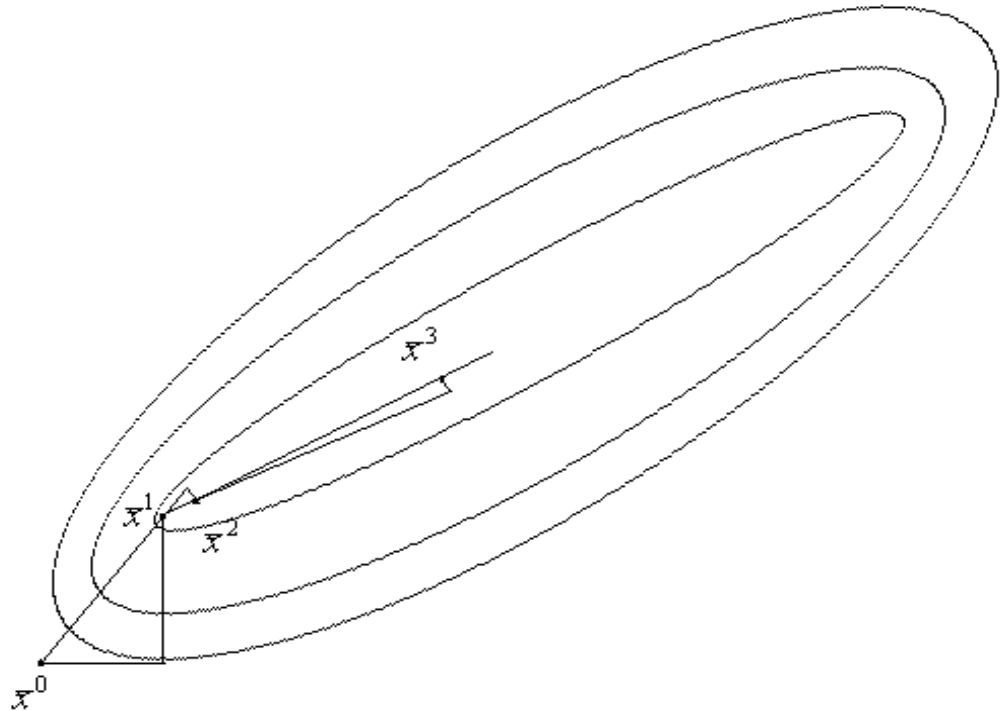


Рис. 3.

Для работы алгоритма необх., чт. ни один из векторов системы $\bar{S}_1^0, \bar{S}_2^0, \dots, \bar{S}_n^0$ не стал нулевым вектором. Для этого в алгоритме следует располагать параметры $\lambda_1^k, \lambda_2^k, \dots, \lambda_n^k$ в порядке убывания по абсолютному значению, т.е. $|\lambda_1^k| > |\lambda_2^k| > \dots > |\lambda_n^k|$. Тогда ес. любые m из λ_i^k обращаются в нуль, то отыскиваются новые направления по (1) только для тех $(n - m)$ направлений, для которых $\lambda_i^k \neq 0$, оставшиеся же m направлений остаются неизменными: $\bar{S}_i^{k+1} = \bar{S}_i^k$, $i = \overline{(n - m + 1), n}$. Т.к. первые $(n - m)$ векторов взаимно ортогональны, $\lambda_i^k = 0$, $i = \overline{(n - m + 1), n}$, первые $(n - m)$ векторов не будут иметь составляющих в направлениях \bar{S}_i^{k+1} , $i = \overline{(n - m + 1), n}$. А поскольку эти последние направления взаимно ортогональны, то из этого следует, что все направления являются взаимно ортогональными.

Палмером было показано, что \bar{B}_{j+1}^k и $\|\bar{B}_{j+1}^k\|$ пропорциональны λ_j^k (при условии, что $\sum_{i=j}^n (\lambda_i^k)^2 \neq 0$). \Rightarrow , при вычислении $\bar{S}_j^{k+1} = \bar{B}_j^k / \|\bar{B}_j^k\|$, величина

λ_j^k сокращается, и, т.о., \bar{S}_j^{k+1} остается определенным, ес. даже $\lambda_j^k = 0$.

Имея это в виду, Палмер предложил для вычисления \bar{S}_j^{k+1} следующие соотношения:

$$\bar{A}_i^k = \sum_{j=i}^n \lambda_j^k \cdot \bar{S}_j^k,$$

$$\bar{S}_i^{k+1} = \frac{\bar{A}_i^k \cdot \|\bar{A}_{i-1}^k\|^2 - \bar{A}_{i-1}^k \cdot \|\bar{A}_i^k\|^2}{\|\bar{A}_{i-1}^k\| \cdot \|\bar{A}_i^k\| \cdot [\|\bar{A}_{i-1}^k\|^2 - \|\bar{A}_i^k\|^2]^{1/2}}, \quad i = 2, \dots, n, \quad \bar{S}_1^{k+1} = \frac{\bar{A}_1^k}{\|\bar{A}_1^k\|}.$$

Критерии останова алгоритма могут быть стандартными (т.е. описанными в предыдущих алгоритмах прямых методов).

2.Метод по=>го уточнения оценок.

Иногда называют еще двойственным симплекс-методом. Ранее говорилось, что одновременно с решением прямой задачи, решается и двойственная задача. Ес. проследить за получающимися преобразованиями двойственной таблицы и переменных u_i и x_j , записав таблицу для двойственной задачи в обычном виде, то получим описание нового метода – метода по=>го уточнения оценок.

Пусть дана задача:

Симплексная таблица, построенная для данной задачи, будет иметь вид:

В методе по=>го уточнения оценок сначала избавляются от отрицательности в W -строке (получают псевдоплан), а затем, перебирая псевдопланы, ищут оптимальный план (первый найденный опорный).

Правило выбора разрешающего элемента для избавления от отрицательности в W -строке.

- 1) Ес. все коэффициенты W -строки неотрицательны, то 0 явл-ся оценкой снизу для целевой ф-ции W и можно переходить к отысканию оптимального решения. Иначе выбираем некоторый $b_r < 0$ и рассматриваем r -й столбец.
- 2) Находим в r -м столбце какой-нибудь из отрицательных элементов, например, $a_{rs} < 0$. Тогда строку с номером s , содержащую a_{rs} , выбираем в качестве разрешающей строки. Ес. все коэффициенты r -го столбца неотрицательны, то либо W неограничена снизу ($W \rightarrow -\infty$), либо система ограничений противоречива. (Из противоречивости двойственной не следует неограниченность прямой задачи).

- 3) Находим неотрицательные отношения коэффициентов W -строки к коэффициентам разрешающей (s -й) строки. В качестве разрешающего берем тот элемент разрешающей s -й строки, для которого это отношение положительно и минимально, т.е. выбираем некоторый коэффициент a_{sk} , для которого

$$\frac{b_k}{a_{sk}} = \min_j \left\{ \frac{b_j}{a_{sj}} \mid \frac{b_j}{a_{sj}} > 0 \right\}.$$

Выбрав разрешающий элемент, делаем шаг обыкновенного жорданова исключения. Указанная по-=>сть действий выполняется до тех пор, пока все коэффициенты W -строки не станут неотрицательными. Например, будет получена следующая таблица

	v_1	v_s	u_{s+1}	\cdots	u_m	1
u_1 =	b_{11}	b_{s1}	$b_{s+1,1}$	\cdots	b_{m1}	q_1
....	\cdots	\cdots
u_s =	b_{1s}	b_{ss}	$b_{s+1,s}$	\cdots	b_{ms}	q_s
v_{s+1} =	$b_{1,s+1}$	$b_{s,s+1}$	$b_{s+1,s+1}$	\cdots	$b_{m,s+1}$	q_{s+1}
....	\cdots	\cdots
v_n =	$b_{1,n}$	$b_{s,n}$	$b_{s+1,n}$	\cdots	b_{mn}	q_n
W =	b_1	b_s	b_{s+1}	\cdots	b_m	q_0

Ес. все q_1, \dots, q_n – неотрицательны, то таблица соответствует оптимальному решению и $q_0 = \min W$, иначе, q_0 – оценка снизу для W .

Правило выбора разрешающего элемента при поиске оптимального решения.

- 1) В качестве разрешающей строки берем строку, содержащую отрицательный коэффициент, например, $q_s < 0$, и строка с номером s будет разрешающей.
- 2) В качестве разрешающего выбираем тот положительный коэффициент b_{ks} строки s , для которого

$$\frac{b_k}{b_{sk}} = \min_j \left\{ \frac{b_j}{b_{sj}} \mid \frac{b_j}{b_{sj}} > 0 \right\}.$$

Ес. в строке с номером s нет положительных коэффициентов, то *ограничения задачи противоречивы*.

После выбора разрешающего элемента делаем шаг обыкновенного жорданова исключения.

После конечного числа шагов либо найдем оптимальное решение, либо можно убедимся в противоречивости ограничений задачи.

Замечание: Ес. в симплекс-методе мы приближаемся к оптимальному решению при поиске минимума *сверху*, передвигаясь по опорным планам, то в методе по=>го уточнения оценок при поиске минимума к оптимальному решению приближаемся *снизу*, причем промежуточные планы (*псевдопланы*) не являются опорными (лежат вне многогранника решений). Первое же допустимое решение (опорный план) будет оптимальным.

4 билет

1.Симплексный метод Нелдера и Мида.

В процессе поиска осуществляется работа с регулярными симплексами.

Регулярные многогранники в пространстве E^n называются **симплексами**. Для $n = 2$ регулярный симплекс представляет собой равносторонний треугольник; при $n = 3$ - тетраэдр и т.д.

Координаты вершин регулярного симплекса в n -мерном пространстве могут быть определены следующей матрицей D , в которой столбцы представляют собой вершины симплекса, пронумерованные от 1 до $(n + 1)$, а строки – координаты вершин, $i = \overline{1, n}$. Матрица имеет размерность $n \times (n + 1)$:

$$D = \begin{bmatrix} 0 & d_1 & d_2 & \dots & d_2 \\ 0 & d_2 & d_1 & \dots & d_2 \\ 0 & d_2 & d_2 & \dots & d_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & d_2 & d_2 & \dots & d_1 \end{bmatrix}_{n \times (n+1)},$$

где:

$$d_1 = \frac{t}{n\sqrt{2}}(\sqrt{n+1} + n - 1); \quad d_2 = \frac{t}{n\sqrt{2}(\sqrt{n+1} - 1)};$$

t – расстояние м/у вершинами.

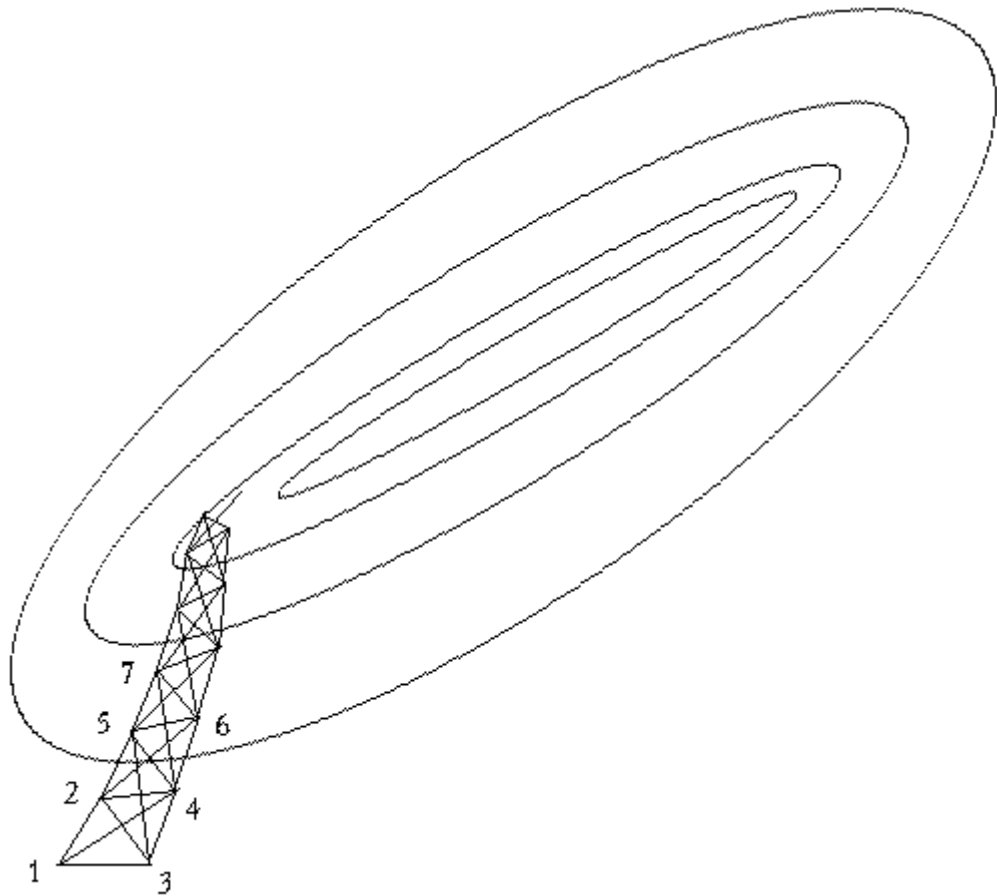


Рис.4.

В самом простом виде симплексный алгоритм заключается в следующем. Строится регулярный симплекс. Из вершины, в которой $f(\bar{x})$ максимальна (т.1) проводится проектирующая прямая ч/з центр тяжести симплекса. Затем т.1 исключается и строится новый *отраженный* симплекс из оставшихся старых точек и одной новой, расположенной на проектирующей прямой на надлежащем расстоянии от центра тяжести.

Продолжение этой процедуры, в которой каждый раз исключается вершина, где целевая ф-ция максимальна, а также использование правил уменьшения размера симплекса и предотвращение циклического движения в окрестности экстремума позволяет достаточно эффективно определять минимум для "хороших" функций. Но для "овражных" функций такой поиск неэффективен.

В симплексном алгоритме Нелдера и Мида минимизация функций n переменных осуществляется с использованием деформируемого многогранника.

Будем рассматривать k -ю итерацию алгоритма. Путь $\bar{x}_i^k = [x_{i1}^k, x_{i2}^k, \dots, x_{in}^k]^T$, $i = 1, \dots, (n+1)$, явл-ся i -й вершиной в E^n на k -м этапе поиска, $k = 0, 1, 2, \dots$, и пусть значения целевой ф-ции в вершине $f(\bar{x}_i^k)$. Отметим

вершины с минимальным и максимальным значениями. И обозначим их следующим образом:

$$f(\bar{x}_h) = \max \{f(\bar{x}_1), \dots, f(\bar{x}_{n+1})\} \text{ и } f(\bar{x}_l) = \min \{f(\bar{x}_1), \dots, f(\bar{x}_{n+1})\}.$$

Многогранник в E^n состоит из $n+1$ вершин $\bar{x}_1^k, \bar{x}_2^k, \dots, \bar{x}_{n+1}^k$. Обозначим ч/з \bar{x}_{n+2}^k - центр тяжести вершин без точки \bar{x}_h^k с максимальным значением ф-ции. Координаты этого центра вычисляются по ф-ле:

$$x_{n+2,j}^k = \frac{1}{n} \left[\sum_{i=1}^{n+1} x_{i,j}^k - x_{h,j}^k \right], \quad j = 1, \dots, n. \quad (1)$$

Начальный многогранник обычно выбирается в виде регулярного симплекса (с вершиной в начале координат). Можно начало координат поместить в центр тяжести. Процедура отыскания вершин в E^n , в которых $f(\bar{x})$ имеет лучшее значение, состоит из следующих операций: 1) отражения; 2) растяжения; 3) сжатия; 4) редукции.

1. Отражение. Отражение – проектирование точки \bar{x}_h^k ч/з центр тяжести \bar{x}_{n+2}^k в соответствии со следующим соотношением:

$$\bar{x}_{n+3}^k = \bar{x}_{n+2}^k + \alpha \cdot (\bar{x}_{n+2}^k - \bar{x}_h^k), \quad (2)$$

где $\alpha > 0$ - коэффициент отражения.

Вычисляем значение ф-ции в найденной точке $f(\bar{x}_{n+3}^k)$. Ес. значение ф-ции в данной точке $f(\bar{x}_{n+3}^k) \geq f(\bar{x}_h^k)$, то переходим к четвертому пункту алгоритма – операции редукции.

Ес. $f(\bar{x}_{n+3}^k) < f(\bar{x}_h^k) \wedge f(\bar{x}_{n+3}^k) < f(\bar{x}_l^k)$, то выполняем операцию растяжения.

В противном случае, ес. $f(\bar{x}_{n+3}^k) < f(\bar{x}_h^k) \wedge f(\bar{x}_{n+3}^k) \geq f(\bar{x}_l^k)$, то выполняется операция сжатия.

2. Растяжение. Эта операция заключается в следующем. Ес.

$f(\bar{x}_{n+3}^k) < f(\bar{x}_l^k)$ (меньше минимального значения на k -м этапе), то вектор $(\bar{x}_{n+3}^k - \bar{x}_{n+2}^k)$ растягивается в соответствии с соотношением

$$\bar{x}_{n+4}^k = \bar{x}_{n+2}^k + \gamma \cdot (\bar{x}_{n+3}^k - \bar{x}_{n+2}^k), \quad (3)$$

где $\gamma > 0$ - коэффициент растяжения. Ес. $f(\bar{x}_{n+4}^k) < f(\bar{x}_l^k)$, то \bar{x}_l^k заменяется на \bar{x}_{n+4}^k и процедура продолжается с операции 1) при $k = k + 1$.

В противном случае \bar{x}_l^k заменяется на \bar{x}_{n+3}^k и переходим к операции отражения 1).

3. Сжатие. Ес. $f(\bar{x}_{n+3}^k) > f(\bar{x}_i^k)$ для $\forall i \ i \neq h$, то вектор $(\bar{x}_h^k - \bar{x}_{n+2}^k)$ сжимается в соответствии с ф-лой

$$\bar{x}_{n+5}^k = \bar{x}_{n+2}^k + \beta \cdot (\bar{x}_h^k - \bar{x}_{n+2}^k),$$

где $0 < \beta < 1$ - коэффициент сжатия. После этого, точка \bar{x}_h^k заменяется на \bar{x}_{n+5}^k , и переходим к операции отражения 1) с $k = k + 1$. Заново ищется \bar{x}_h^{k+1} .

4. Редукция. Ес. $f(\bar{x}_{n+3}^k) > f(\bar{x}_h^k)$, то все векторы $(\bar{x}_i^k - \bar{x}_l^k)$, где $i = \overline{1, (n+1)}$ уменьшаются в два раза с отсчетом от точки \bar{x}_l^k по ф-ле

$$\bar{x}_i^k = \bar{x}_l^k + 0.5 \cdot (\bar{x}_i^k - \bar{x}_l^k), \quad i = \overline{1, (n+1)}$$

и переходим к операции отражения (на начало алгоритма с $k = k + 1$).

В качестве критерия останова могут быть взяты те же критерии, что и в остальных алгоритмах. Можно также использовать критерий останова

следующего вида: $\left\{ \frac{1}{n+1} \cdot \sum_{i=1}^{n+1} [f(\bar{x}_i^k) - f(\bar{x}_{n+2}^k)]^2 \right\}^{1/2} < \varepsilon$.

Выбор коэффициентов α, β, γ обычно осуществляется эмпирически. После того как многогранник подходящим образом промасштабирован, его размеры должны поддерживаться неизменными пока изменения в топологии задачи не потребуют многогранника другой формы. Чаще всего выбирают $\alpha = 1$, $0.4 \leq \beta \leq 0.6$, $2 \leq \gamma \leq 3$.

2. Транспортная задача. Определение опорного решения.

Транспортная задача линейного программирования формулируется следующим образом. Необходимо минимизировать транспортные расходы

$$Q(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij} \rightarrow \min \quad \text{при ограничениях}$$

$$\left. \begin{aligned} \sum_{i=1}^m x_{ij} &= b_j, & j &= \overline{1, n}, \\ \sum_{j=1}^n x_{ij} &= a_i, & i &= \overline{1, m}, \\ x_{ij} &\geq 0, & i &= \overline{1, m}, \quad j = \overline{1, n}, \end{aligned} \right\},$$

где c_{ij} - стоимость перевозки единицы продукции из пункта i в пункт j ; x_{ij} – планируемая величина перевозок из пункта i в пункт j (план перевозок X – матрица размерности $m \times n$); b_j - потребности в продукте в пункте j ; a_i - запасы в пункте i .

Предполагается, что модель *закрытого* типа, то есть $\sum_{j=1}^n b_j = \sum_{i=1}^m a_i$.

Если модель открытого типа $\left(\sum_{j=1}^n b_j \neq \sum_{i=1}^m a_i \right)$, то ее всегда можно привести

к закрытому типу введением фиктивного пункта производства или фиктивного пункта потребления:

1. Если $\sum_{j=1}^n b_j < \sum_{i=1}^m a_i$, то $b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$, тогда $\sum_{j=1}^{n+1} b_j = \sum_{i=1}^m a_i$, причем $c_{i,n+1} = 0 \quad \forall i$.
2. Если $\sum_{j=1}^n b_j > \sum_{i=1}^m a_i$, то $a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$, $\sum_{j=1}^n b_j = \sum_{i=1}^{m+1} a_i$ и $c_{m+1,j} = 0 \quad \forall j$.

Транспортная задача представляет собой задачу линейного программирования и, естественно, ее можно решить с использованием метода последовательного улучшения плана или метода последовательного уточнения оценок. В этом случае основная трудность бывает связана с числом переменных задачи ($m \times n$) и числом ограничений ($m+n$). Поэтому специальные алгоритмы оказываются более эффективными. К таким алгоритмам относятся *метод потенциалов* и *венгерский метод*.

Алгоритм метода потенциалов, его называют еще модифицированным распределительным алгоритмом, начинает работу с некоторого опорного плана транспортной задачи (допустимого плана перевозок). Для построения опорного плана обычно используется один из двух методов: *метод северо-западного угла* или *метод минимального элемента*.

Метод северо-западного угла

Он позволяет найти некоторый допустимый план перевозок. Составим транспортную таблицу некоторой задачи.

	b_j				
	30	80	20	30	90
a_i					

120	2 30	4 80	2 10	3 —	8 —
30	3 	5 	6 10	6 20	2 —
40	6 	8 	7 	4 10	5 30
60	3	4	2 	1 	4 60

В данном случае, имеем задачу закрытого типа, т.к. $\sum_{i=1}^4 a_i = 250 = \sum_{j=1}^5 b_j$.

При построении плана должны учитывать, что сумма перевозок в столбце должна оказаться равной потребностям в данном пункте, а сумма перевозок в строке запаса в пункте, соответствующем данной строке.

Заполнение начинается с верхнего левого угла таблицы. Величина перевозки устанавливается равной минимальной из величин: величины остатка запасов в пункте i или величины еще неудовлетворенного спроса в пункте j .

- Если ресурс в данной строке исчерпан, то переходим к перевозке в следующей строке текущего столбца (на одну строку вниз).
- Если потребности для данного пункта (столбца) удовлетворены, то переходим к следующей перевозке текущей строки в следующем столбце.

Затраты на перевозку по построенному плану равны:

$$Q = 30 \times 2 + 4 \times 80 + 2 \times 10 + 6 \times 10 + 6 \times 20 + 4 \times 10 + 5 \times 30 + 4 \times 60 = 1010.$$

Естественно, что найденный план далек от оптимального.

Метод минимального элемента

В таблице отыскивается $\min \{c_{ij}\}$ и в первую очередь заполняется соответствующая клетка: $x_{ij} = \min \{a_i, b_j\}$. Затем вычеркивается остаток соответствующей строки, если $a_i < b_j$, или столбца, если $a_i > b_j$, и корректируем остатки запасов и неудовлетворенного спроса. В оставшихся клетках таблицы снова отыскивается минимальная стоимость перевозки и заполняется соответствующая клетка и т.д.

	b_j				
	30	80	20	30	90
a_i					

120	2 30	4 80	2 	3 	8 10
30	3 	5 —	6 	6 	2 30
40	6 	8 	7	4	5 40
60	3	4	2 20	1 30	4 10

Затраты на перевозку по построенному плану равны:

$$Q = 30 \times 2 + 4 \times 80 + 8 \times 10 + 2 \times 30 + 5 \times 40 + 2 \times 20 + 1 \times 30 + 4 \times 10 = 830 .$$

Этот план лучше, но утверждать, что он оптимален, нельзя.

Определение 1. Набором называется произвольная совокупность перевозок транспортной таблицы.

Определение 2. Цепью называют такие наборы, когда каждая пара соседних клеток в цепи расположены либо в одном столбце, либо в одной строке.

Определение 3. Циклом называется цепь, крайние элементы которой находятся либо в одной строке, либо в одном столбце.

5 билет

1.Сопряженные направления.

Метод Пауэлла относится к прямым методам (методам нулевого порядка). Этим методом наиболее эффективно осуществляется минимизация функций, близких к квадратичным. На каждой итерации алгоритма поиск осуществляется вдоль системы сопряженных направлений.

Два направления поиска \bar{S}_i, \bar{S}_j называются **сопряженными**, если

$$\bar{S}_i^T \cdot H \cdot \bar{S}_j = 0, \quad i \neq j,$$

$$\bar{S}_i^T \cdot H \cdot \bar{S}_i \geq 0, \quad i = j,$$

где H - положительно определенная квадратная матрица.

Обоснование применения сопряженных направлений в алгоритмах оптимизации. В методе Пауэлла $H = \nabla^2 f(\bar{x}^k)$ - матрица вторых частных производных. Идеи метода Пауэлла относятся к квадратичной функции $f(\bar{x})$.

Основная идея заключается в том, что если на каждом этапе поиска определяется минимум квадратичной функции $f(\bar{x})$ вдоль каждого из p

$(p < n)$ - сопряженных направлений и если затем в каждом из направлений делается шаг до минимальной точки, то полное перемещение от начала до шага с номером p сопряжено ко всем поднаправлениям поиска.

Идея использования сопряженных направлений лежит в основе ряда алгоритмов.

Пусть $f(\bar{x})$ - квадратичная функция и процесс минимизации начинается в точке \bar{x}^0 с начальным направлением \bar{S}^1 . Для удобства возьмем этот вектор единичным, т.е. $(\bar{S}^1)^T \cdot \bar{S}^1 = 1$. Тогда вектор $\bar{x}^1 = \bar{x}^0 + \lambda^1 \cdot \bar{S}^1$ и длина шага λ^1 определяется из условия минимальности функции в данном направлении т.е.

$$\lambda^1 = \arg \min_{\lambda} f(\bar{x}^0 + \lambda \cdot \bar{S}^1).$$

Для квадратичной ф-ции

$$\frac{df(\bar{x}^0 + \lambda \cdot \bar{S}^1)}{d\lambda} = \nabla^T f(\bar{x}^0) \bar{S}^1 + (\bar{S}^1)^T \nabla^2 f(\bar{x}^0) \lambda \bar{S}^1 = 0, \quad (1)$$

и, т.о., оптимальное значение λ на первом шаге определяется в соответствии с соотношением

$$\lambda^1 = \frac{-\nabla^T f(\bar{x}^0) \bar{S}^1}{(\bar{S}^1)^T \cdot \nabla^2 f(\bar{x}^0) \cdot \bar{S}^1} = -\frac{\nabla^T f(\bar{x}^0) \bar{S}^1}{(\bar{S}^1)^T \cdot H \cdot \bar{S}^1}, \quad (2)$$

где $H = \nabla^2 f(\bar{x}^k)$.

Из точки \bar{x}^1 процесс минимизации должен осуществляться в другом сопряженном направлении \bar{S}^2 и при этом $(\bar{S}^2)^T \cdot H \cdot \bar{S}^1 = 0$.

Отметим, что квадратичная функция может быть представлена в виде

$$f(\bar{x}) = a + b^T \cdot \bar{x} + \frac{1}{2} \bar{x}^T \cdot H \cdot \bar{x},$$

где положительно определенная матрица $H = \nabla^2 f(\bar{x})$.

В общем случае система n линейно независимых направлений поиска $\bar{S}^1, \bar{S}^2, \dots, \bar{S}^n$ называется **сопряженной** по отношению к некоторой положительно определенной матрице H , если

$$(\bar{S}^i)^T \cdot H \cdot \bar{S}^j = 0, \quad 1 \leq i \neq j \leq n.$$

Так как сопряженные направления линейно независимы, то любой вектор в пространстве E^n можно выразить через $\bar{S}^1, \bar{S}^2, \dots, \bar{S}^n$ следующим образом:

$$\left. \begin{aligned} \bar{V} &= \sum_{j=1}^n v^j \cdot \bar{S}^j \\ \text{где} \quad v^j &= \frac{(\bar{S}^j)^T \cdot H \cdot \bar{V}}{(\bar{S}^j)^T \cdot H \cdot \bar{S}^j} \end{aligned} \right\}. \quad (3)$$

Известная формула, примем ее на веру.

Для некоторой матрицы H всегда существует, по крайней мере, одна система из n взаимно сопряженных направлений, так как сами собственные векторы матрицы H представляют собой такую систему.

Отметим, что для квадратичной функции справедливо следующее соотношение, которое потребуется в дальнейшем:

$$H^{-1} = \sum_{j=1}^n \frac{\bar{S}^j \cdot (\bar{S}^j)^T}{(\bar{S}^j)^T \cdot H \cdot \bar{S}^j}. \quad (4)$$

Чтобы убедиться в его справедливости, рассмотрим матрицу $\sum_{j=1}^n \alpha_j \bar{S}^j (\bar{S}^j)^T$.

Умножение ее справа на $H \bar{S}^k$ дает

$$\left[\sum_{j=1}^n \alpha_j \bar{S}^j (\bar{S}^j)^T \right] H \bar{S}^k = \alpha_k \bar{S}^k (\bar{S}^k)^T H \bar{S}^k = \bar{S}^k,$$

если положить $\alpha_k = \left[(\bar{S}^k)^T H \bar{S}^k \right]^{-1}$.

Вообще говоря, справедливо общее правило, заключающееся в том, что если используются сопряженные направления для поиска минимума квадратичной функции $f(\bar{x})$, то эта функция может быть минимизирована за n шагов по одному в каждом из сопряженных направлений. Более того, порядок использования сопряженных направлений несущественен.

Покажем, что это действительно так. Пусть $f(\bar{x})$ - квадратичная функция и

$$f(\bar{x}) = a + \bar{b}^T \cdot \bar{x} + \frac{1}{2} \bar{x}^T \cdot H \cdot \bar{x},$$

при этом $\nabla f(\bar{x}) = \bar{b} + H \cdot \bar{x}$.

В точке минимума $\nabla f(\bar{x}^*) = 0$, и эта точка $\bar{x}^* = -H^{-1} \cdot \bar{b}$.

Заметим, что $\nabla^T f(\bar{x}^k) \cdot \bar{S}^k = (\bar{S}^k)^T \cdot \nabla f(\bar{x}^k)$.

Так как
$$\bar{x}^1 = \bar{x}^0 + \lambda^1 \cdot \bar{S}^1, \quad (5)$$

где λ^1 определяется в соответствии с соотношением (2):

$$\lambda^1 = \frac{-\nabla^T f(\bar{x}^0) \bar{S}^1}{(\bar{S}^1)^T \cdot \nabla^2 f(\bar{x}^0) \cdot \bar{S}^1} = -\frac{\nabla^T f(\bar{x}^0) \bar{S}^1}{(\bar{S}^1)^T \cdot H \cdot \bar{S}^1},$$

затем минимум находится в следующем сопряженном направлении по аналогичным формулам $\bar{x}^2 = \bar{x}^1 + \lambda^2 \cdot \bar{S}^2$ и т.д., то на n -м шаге имеем

$$\bar{x}^n = \bar{x}^0 + \sum_{i=1}^n \lambda^i \cdot \bar{S}^i. \quad (6)$$

На каждом шаге минимизируем функцию $f(\bar{x}^{i-1} + \lambda^i \cdot \bar{S}^i)$ в направлении \bar{S}^i , чт. получить λ^i . На основании (2) это приводит к следующему

выражению
$$\bar{x}^n = \bar{x}^0 - \sum_{i=1}^n \left[\frac{(\bar{S}^i)^T \cdot \nabla f(\bar{x}^{i-1})}{(\bar{S}^i)^T \cdot H \cdot \bar{S}^i} \right] \cdot \bar{S}^i. \quad (7)$$

Кроме того,

$$(\bar{S}^i)^T \cdot \nabla f(\bar{x}^{i-1}) = (\bar{S}^i)^T \cdot (H \cdot \bar{x}^{i-1} + \bar{b}) = (\bar{S}^i)^T \cdot \left\{ H \cdot \left[\bar{x}^0 + \sum_{k=1}^{i-1} \lambda^k \cdot \bar{S}^k \right] + \bar{b} \right\}$$

и получаем
$$(\bar{S}^i)^T \cdot \nabla f(\bar{x}^{i-1}) = (\bar{S}^i)^T \cdot [H \cdot \bar{x}^0 + \bar{b}],$$

так как все $(\bar{S}^i)^T \cdot H \cdot \bar{S}^k = 0, \forall i \neq k$, вследствие сопряженности \bar{S}^i и \bar{S}^k .

Т.о.,
$$\bar{x}^n = \bar{x}^0 - \sum_{i=1}^n \frac{(\bar{S}^i)^T \cdot (H \cdot \bar{x}^0 + \bar{b}) \cdot \bar{S}^i}{(\bar{S}^i)^T \cdot H \cdot \bar{S}^i}. \quad (8)$$

Выразим векторы \bar{x}^0 и $H^{-1} \cdot \bar{b}$ через систему сопряженных векторов \bar{S}^i следующим образом (по аналогии с (3)):

$$\bar{x}^0 = \sum_{i=1}^n \frac{(\bar{S}^i)^T \cdot H \cdot \bar{x}^0 \cdot \bar{S}^i}{(\bar{S}^i)^T \cdot H \cdot \bar{S}^i},$$

$$H^{-1} \cdot \bar{b} = \sum_{i=1}^n \frac{(\bar{S}^i)^T \cdot H \cdot (H^{-1} \cdot \bar{b}) \cdot \bar{S}^i}{(\bar{S}^i)^T \cdot H \cdot \bar{S}^i} = \sum_{i=1}^n \frac{(\bar{S}^i)^T \cdot \bar{b} \cdot \bar{S}^i}{(\bar{S}^i)^T \cdot H \cdot \bar{S}^i}.$$

С учетом этого из (8), получим
$$\bar{x}^n = \bar{x}^0 - \bar{x}^0 - H^{-1} \cdot \bar{b} = -H^{-1} \cdot \bar{b}. \quad (9)$$

Т.о., точка \bar{x}^n , полученная в результате минимизации квадратичной функции на n -м шаге, совпадает с точкой минимума квадратичной функции $f(\bar{x})$. Покажем, что для сопряженных направлений, если $f(\bar{x})$

каждый раз минимизируется в сопряженном направлении \bar{S}^j в соответствии с формулой (2), то при этом выполняется следующее

равенство: $(\bar{S}^j)^T \cdot \nabla f(\bar{x}^l) = 0, \quad 1 \leq j \leq l-1,$

при использовании не более чем n направлений, то есть $\nabla f(\bar{x}^l)$ ортогонален использованным сопряженным направлениям.

Для квадратичной функции $\nabla f(\bar{x}^l) = \bar{b} + H \cdot \bar{x}^l$. Следовательно,

$$\nabla f(\bar{x}^l) = \bar{b} + H \cdot \left[\bar{x}^k + \sum_{j=k}^{l-1} \lambda^j \cdot \bar{S}^j \right] = \bar{b} + H \bar{x}^k + H \sum_{j=k}^{l-1} \lambda^j \cdot \bar{S}^j,$$

где \bar{x}^k - произвольная точка, из которой начинается поиск по сопряженным направлениям.

Поскольку $\nabla f(\bar{x}^k) = \bar{b} + H \cdot \bar{x}^k$, то $\nabla f(\bar{x}^l) = \nabla f(\bar{x}^k) + \sum_{j=k}^{l-1} \lambda^j \cdot H \cdot \bar{S}^j$.

Умножение этого уравнения слева на $(\bar{S}^{k-1})^T$ дает

$$(\bar{S}^{k-1})^T \cdot \nabla f(\bar{x}^l) = (\bar{S}^{k-1})^T \cdot \nabla f(\bar{x}^k) + \sum_{j=k}^{l-1} \lambda^j \cdot (\bar{S}^{k-1})^T \cdot H \cdot \bar{S}^j.$$

Первый член в правой части $(\bar{S}^{k-1})^T \cdot \nabla f(\bar{x}^k) = 0$, так как градиент в точке \bar{x}^k ортогонален направлению предыдущего спуска, если точка получена в результате минимизации функции в этом направлении. Кроме того, все остальные слагаемые под знаком суммы исчезают вследствие сопряженности направлений \bar{S}^{k-1} и \bar{S}^j , и т.о.

$$(\bar{S}^j)^T \cdot \nabla f(\bar{x}^l) = 0, \quad 1 \leq j \leq l-1. \quad (10)$$

То есть, градиент в точке \bar{x}^l ортогонален ко всем использованным ранее сопряженным направлениям.

2.Метод потенциалов для решения транспортной задачи линейного программирования.

Метод позволяет находить оптимальный план перевозок транспортной таблицы. В основе лежит следующая теорема.

Теорема. Для того, чт. некоторый план $X = [x_{ij}]_{m \times n}$ транспортной задачи был оптимальным, необходимо и достаточно, чт. ему соответствовала такая система $m+n$ чисел $u_1, u_2, \dots, u_m; v_1, v_2, \dots, v_n$, для которой выполняются условия:

$$v_j - u_i \leq c_{ij}, \quad i = \overline{1, m}, \quad j = \overline{1, n}, \quad (1)$$

$$v_j - u_i = c_{ij}, \quad \forall x_{ij} > 0. \quad (2)$$

u_i и v_j называются потенциалами соответствующих пунктов отправления и пунктов назначения. Условия (1)-(2) называются условиями потенциальности.

План X будем называть потенциальным, если для него существует система u_i и v_j , удовлетворяющая (1)-(2). Тогда теорема коротко формулируется следующим образом.

Теорема. Для оптимальности плана транспортной задачи необходимо и достаточно, чт. он был потенциален.

Док-во:

Достаточность. Пусть план X потенциален, так что существует система u_i и v_j , удовлетворяющая (1)-(2). Тогда для любого допустимого плана $X' = [x'_{ij}]_{m \times n}$

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x'_{ij} &\geq \sum_{i=1}^m \sum_{j=1}^n (v_j - u_i) x'_{ij} = \sum_{j=1}^n v_j \sum_{i=1}^m x'_{ij} - \sum_{i=1}^m u_i \sum_{j=1}^n x'_{ij} = \\ &= \sum_{j=1}^n v_j b_j - \sum_{i=1}^m u_i a_i = \text{для потенциального плана} \\ &= \sum_{j=1}^n v_j \sum_{i=1}^m x_{ij} - \sum_{i=1}^m u_i \sum_{j=1}^n x_{ij} = \\ &= \sum_{j=1}^n \sum_{i=1}^m v_j x_{ij} - \sum_{i=1}^m \sum_{j=1}^n u_i x_{ij} = \sum_{j=1}^n \sum_{i=1}^m (v_j - u_i) x_{ij} = \sum_{j=1}^n \sum_{i=1}^m c_{ij} x_{ij}, \end{aligned}$$

т.е. стоимость перевозок по любому плану X' не меньше стоимости перевозок по потенциальному плану X . Следовательно, план X оптимален.

Необходимость. Будем рассматривать транспортную задачу, как задачу линейного программирования с минимизацией линейной формы

$$Q(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij} \rightarrow \min$$

при соответствующих ограничениях. Заполним симплексную таблицу и рассмотрим двойственную к ней задачу, что легко получить из таблицы. Прямую таблицу будем заполнять, повернув.

	0=	...	0=	...	0=	0=	...	0=	...	0=	$Q =$
	$-u_1$		$-u_i$		$-u_m$	$-v_1$		$-v_j$		$-v_n$	1
$x_{11} y_{11} =$	-1	...	0	...	0	1	...	0	...	0	c_{11}
...
$x_{1n} y_{1n} =$	-1	...	0	...	0	0	...	0	...	1	c_{1n}
...
$x_{i1} y_{i1} =$	0	...	-1	...	0	1	...	0	...	0	c_{i1}
...
$x_{ij} y_{ij} =$	0	...	-1	...	0	0	...	1	...	0	c_{ij}
...
$x_{in} y_{in} =$	0	...	-1	...	0	0	...	0	...	1	c_{in}
...
$x_{m1} y_{m1} =$	0	...	0	...	-1	1	...	0	...	0	C_{m1}
...
$x_{mn} y_{mn} =$	0	...	0	...	-1	0	...	0	...	1	C_{mn}
1 $w =$	a_1	...	a_i	...	a_n	b_1	...	b_j	...	b_n	0

(Прямую задачу заполняем – как для решения задачи методом последовательного уточнения оценок, с обыкновенными жордановыми исключениями, двойственную – как для решения симплекс-методом).

Получ, что двойств задача им вид: $w = \sum_{j=1}^n b_j v_j - \sum_{i=1}^m a_i u_i \rightarrow \max$ при огранич-х

$$y_{ij} = u_i - v_j + c_{ij} \geq 0, \quad i = \overline{1, m}, \quad j = \overline{1, n}, \quad \text{т.е. } v_j - u_i \leq c_{ij}, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Пусть $X = [x_{ij}]_{m \times n}$ – оптимальное решение транспортной задачи. Тогда на основании первой теоремы двойственности двойственная задача имеет оптимальное решение $u_1^*, \dots, u_m^*; v_1^*, \dots, v_n^*$.

Убедимся, что эти числа являются потенциалами соответствующих пунктов транспортной задачи. Действительно, все u_i^*, v_j^* как опорное решение двойственной задачи удовлетворяют неравенствам (1).

Если $x_{ij}^* > 0$, то по второй теореме двойственности соответствующее ограничение двойственной задачи $y_{ij}^* = u_i^* - v_j^* + c_{ij} \geq 0$ обращается в строгое равенство $v_j^* - u_i^* = c_{ij}$.
Теорема доказана.

Алгоритм метода потенциалов

Алгоритм метода потенциалов состоит из предварительного этапа и повторяющегося основного этапа.

Предварительный этап.

1. Каким-либо способом ищется допустимый план X (методом северо-западного угла или минимального элемента).

2. Для полученного плана строится система $m+n$ чисел $u_1, \dots, u_m, v_1, \dots, v_n$, таких, что $v_j - u_i = c_{ij}, \forall x_{ij} > 0$.
3. Построенная система u_i и v_j исследуется на потенциальность, то есть, план X исследуется на оптимальность. Для этого проверяется $v_j - u_i \leq c_{ij}, \forall x_{ij} = 0$.

Если система не потенциальна, то переходят к основному этапу (т.к. план не оптимален), иначе оптимальный план найден.

Основной этап.

1. Улучшаем план, то есть от плана X переходим к плану X' такому, что $Q(X) \geq Q(X')$.
2. Для плана X' строим новую систему $u'_i, v'_j, i = \overline{1, m}, j = \overline{1, n}$, такую, что $v'_j - u'_i = c_{ij}, \forall x_{ij} > 0$.
3. Исследуем систему u'_i, v'_j на потенциальность. Если система не потенциальна, то переходим на п.1. Иначе найден оптимальный план.

6 билет

1.Алгоритм Пауэлла.

Переход из точки \bar{x}_0^k в точку \bar{x}_n^k на k -м шаге алг-ма Пауэлла осущ-ся в соотв. с ф-лой: $\bar{x}_n^k = \bar{x}_0^k + \sum_{j=1}^n \lambda_j^k \cdot \bar{S}_j^k$. При этом по=> осуществляется

минимизация исходной ф-ции по сопряженным направлениям $\bar{S}_1^k, \dots, \bar{S}_n^k$. Результатом минимизации по каждому из сопряженных направлений явл-ся система параметров $\lambda_1^k, \dots, \lambda_n^k$, при которых ф-ция минимальна в каждом из сопряж-х направл-й: $\lambda_j^k = \arg \min_{\lambda} f(\bar{x}_{j-1}^k + \lambda \bar{S}_j^k), \quad \bar{x}_j^k = \bar{x}_{j-1}^k + \lambda_j^k \bar{S}_j^k$.

Начальную систему сопряженных направлений можно выбрать параллельной осям системы координат. В конце каждой итерации алгоритма Пауэлла необх. выбрать новую сис-му сопряж-х направлений, т.к. ес. этого не сделать, то получим простой покоординатный поиск. В основе построения новой сис-мы лежит следующая теорема.

Теорема: Ес. при начальной точке \bar{x}^0 поиска в направлении вектора \bar{S} минимум ф-ции $f(\bar{x})$ находится к точке \bar{x}^a , а при начальной точке $\bar{x}^1 \neq \bar{x}^0$ поиск минимума ф-ции $f(\bar{x})$ в том же направл-и \bar{S} приводит к точке \bar{x}^b , то при $f(\bar{x}^b) < f(\bar{x}^a)$ направл-е $\bar{x}^b - \bar{x}^a$ сопряжено с направ-ем поиска \bar{S} .

Док-во. Исп-уя ранее получ.рез-ты(10),можно записать,что в первом случае

$$\bar{S}^T \cdot \nabla f(\bar{x}^a) = \bar{S}^T \cdot (H \cdot \bar{x}^a + \bar{b}) = 0,$$

аналогично, во втором случае можно записать

$$\bar{S}^T \cdot \nabla f(\bar{x}^b) = \bar{S}^T \cdot (H \cdot \bar{x}^b + \bar{b}) = 0.$$

Вычитая из второго выражения первое, получим, что

$$\bar{S}^T \cdot H \cdot (\bar{x}^b - \bar{x}^a) = 0.$$

=>, векторы \bar{S} и $(\bar{x}^b - \bar{x}^a)$ являются сопряженными.

Эта теорема непосредственно м.б. распространена на случай нескольких сопряженных направлений следующим образом. Ес., начиная из точки \bar{x}^0 , точка \bar{x}^a определяется после использования при минимизации нескольких сопряженных направлений p ($p < n$). И, аналогично, ес. из точки $\bar{x}^1 \neq \bar{x}^0$ точка \bar{x}^b определяется после использования тех же направлений и ф-ция $f(\bar{x})$ минимизируется на каждом шаге, то вектор $(\bar{x}^b - \bar{x}^a)$ сопряжен ко всем p направлениям.

Следующий рисунок служит иллюстрацией теоремы.

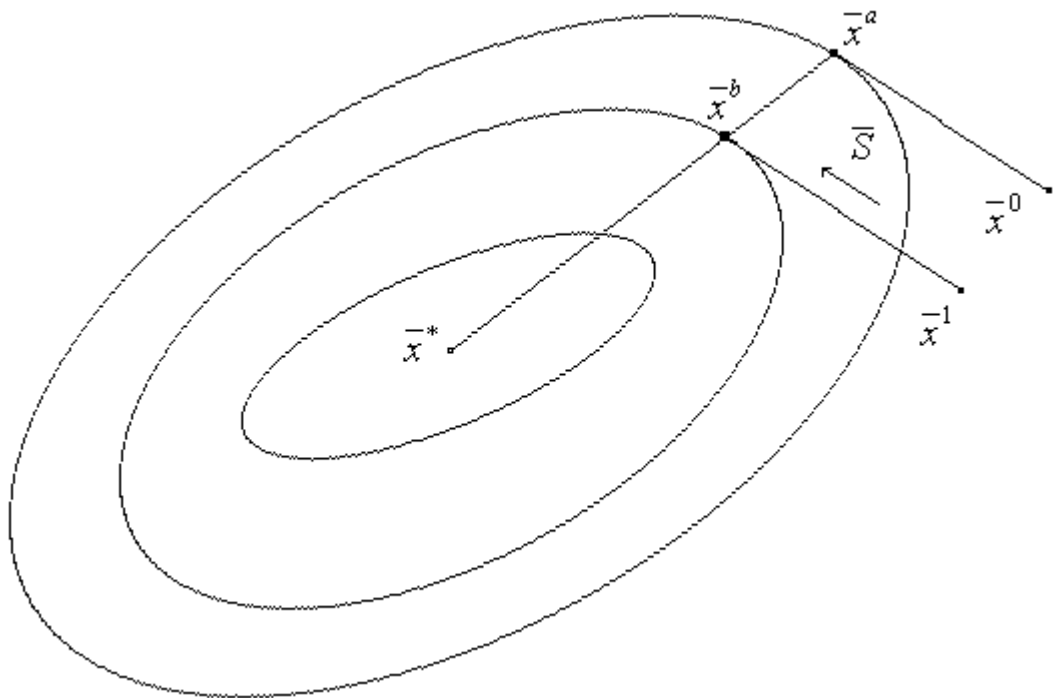


Рис. 5.

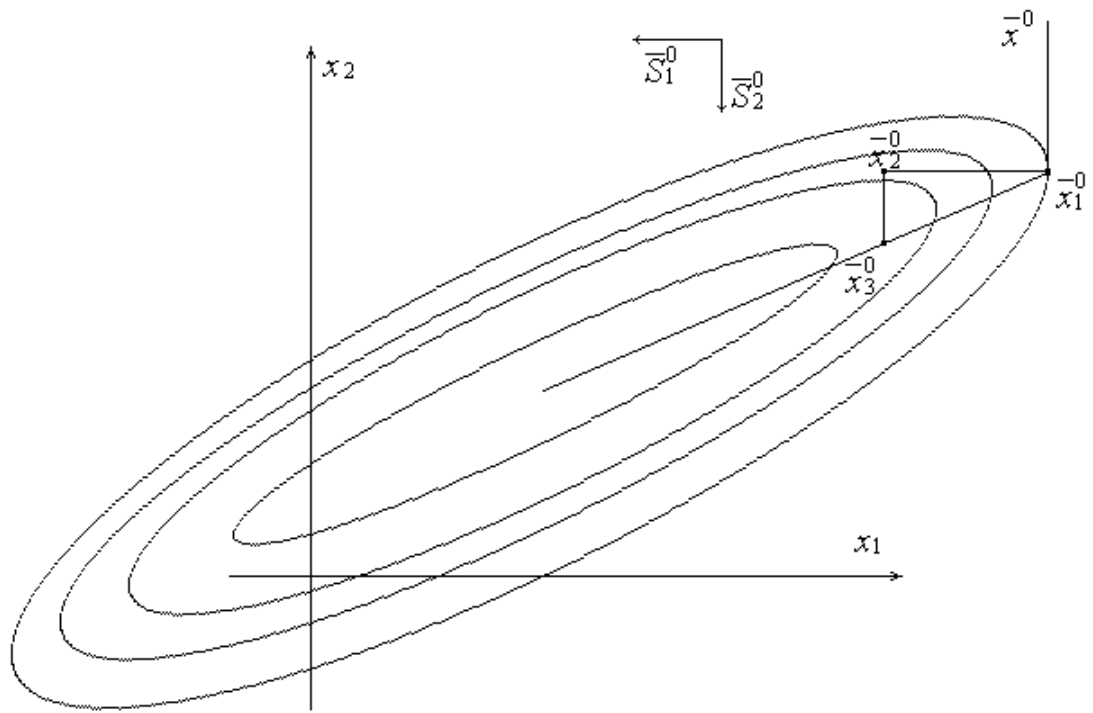


Рис. 6.

Пусть в начальный момент для двумерной задачи поиск осуществляется из точки \bar{x}^0 вдоль направлений, параллельных осям координат: \bar{S}_1^0 и \bar{S}_2^0 .

По \Rightarrow были найдены точки \bar{x}_1^0 , \bar{x}_2^0 , \bar{x}_3^0 (см. рис. 6).

Т.о., определили 2 сопряженных направления, в которых следует вести поиск: \bar{S}_2^0 и $(\bar{x}_3^0 - \bar{x}_1^0)$. В системе исходных направлений \bar{S}_1^0 должно быть заменено на $(\bar{x}_3^0 - \bar{x}_1^0)$, представляющее собой полное перемещение из первого минимума. Направления поиска на следующем этапе:

$$\begin{aligned}\bar{S}_1^1 &= \bar{S}_2^0, \\ \bar{S}_2^1 &= \bar{x}_3^0 - \bar{x}_1^0.\end{aligned}$$

Второй этап начинается с минимизации вдоль направления \bar{S}_2^1 , затем, ес. необх., перемещение в направлении \bar{S}_1^1 . Но в случае квадратичной ф-ции двух переменных после минимизации по двум сопряженным направлениям будет достигнута точка минимума.

В общем случае, на k -м шаге алгоритма Пауэлла используется n линейно независимых направлений поиска. Поиск начинается с точки \bar{x}_0^k и осуществляется по следующему алгоритму:

1. Начиная с точки \bar{x}_0^k , решается по \Rightarrow сть задач минимизации ф-ции $\min_{\lambda} f(\bar{x}_{j-1}^k + \lambda \bar{S}_j^k)$, $j = \overline{1, n}$, в направлениях $\bar{S}_1^k, \dots, \bar{S}_n^k$. При этом находятся

точки $\bar{x}_1^k, \dots, \bar{x}_n^k$, которые минимизируют исходную функцию в заданных направлениях, причем $\bar{x}_1^k = \bar{x}_0^k + \lambda_1 \cdot \bar{S}_1^k$, $\bar{x}_2^k = \bar{x}_1^k + \lambda_2 \cdot \bar{S}_2^k$, ..., $\bar{x}_n^k = \bar{x}_{n-1}^k + \lambda_n \cdot \bar{S}_n^k$.

2. Поиск, осуществляемый на первом этапе, может привести к линейно зависимым направлениям, ес., например, в одном из направлений \bar{S}^i не удастся найти меньшего значения ф-ции. Вследствие этого 2 направления могут стать коллинеарными. Отсюда вытекает, что в системе сопряженных направлений не следует заменять старое направление на новое, ес. после такой замены направления нового набора становятся линейно зависимыми.

На примере квадратичной ф-ции Пауэллом было показано, что при нормировании направлений поиска в соответствии с соотношением:

$$(\bar{S}_i^k)^T \cdot H \cdot \bar{S}_i^k = 1, i = \overline{1, n},$$

определитель матрицы, столбцы которой представляют собой направления поиска, принимает максимальное значение тогда и только тогда, когда \bar{S}_i^k взаимно сопряжены относительно матрицы H . Он пришел к выводу, что направление полного перемещения на k -м шаге должно заменять предыдущее направление только в том случае, когда заменяющий вектор увеличивает определитель матрицы направлений поиска. Т.к. только тогда новый набор направлений будет более эффективным.

Для такой проверки из точки \bar{x}_n^k делается дополнительный шаг в направлении $(\bar{x}_n^k - \bar{x}_0^k)$, соответствующий полному перемещению на k -м этапе и получают точку $(2\bar{x}_n^k - \bar{x}_0^k)$. Для проверки того, что определитель матрицы направлений поиска увеличивается при включении нового направления, делается шаг 3.

3. Обозначим наибольшее уменьшение $f(\bar{x})$ на k -м шаге

$$\Delta^k = \max_{i=\overline{1, n}} \{f(\bar{x}_{i-1}^k) - f(\bar{x}_i^k)\},$$

соответствующее направление поиска обозначим ч/з \bar{S}_m^k .

Обозначим: $f_1 = f(\bar{x}_0^k)$, $f_2 = f(\bar{x}_n^k)$, $f_3 = f(2\bar{x}_n^k - \bar{x}_0^k)$, где

$$\bar{x}_0^k = \bar{x}_n^{k-1}, \quad \bar{x}_n^k = \bar{x}_{n-1}^k + \lambda_n \bar{S}_n^k = \bar{x}_0^k + \sum_{i=1}^n \lambda_i \bar{S}_i^k.$$

Тогда, ес. $f_3 \geq f_1$ и (или) $(f_1 - 2f_2 + f_3)(f_1 - f_2 - \Delta^k)^2 \geq 0.5\Delta^k(f_1 - f_3)^2$, то следует использовать на $(k+1)$ -м этапе те же направления $\bar{S}_1^k, \dots, \bar{S}_n^k$, что и на k -м этапе, то есть $\bar{S}_i^{k+1} = \bar{S}_i^k$, $i = \overline{1, n}$, и начать поиск из точки

$\bar{x}_0^{k+1} = \bar{x}_n^k$ или из точки $\bar{x}_0^{k+1} = 2\bar{x}_n^k - \bar{x}_0^k = \bar{x}_{n+1}^k$, в зависимости от того, в какой точке f -ция принимает минимальное значение.

4. Ес. тест на шаге 3 не прошел, то ищется минимум $f(\bar{x})$ в направлении вектора \bar{S}_{n+1}^k , проведенного из \bar{x}_0^k в \bar{x}_n^k : $\bar{S}_{n+1}^k = (\bar{x}_n^k - \bar{x}_0^k)$. Точка этого минимума берется в качестве начальной точки на $(k+1)$ -м этапе. А в системе сопряженных направлений сохраняются все, кроме направления \bar{S}_m^k , которое заменяется на новое направление \bar{S}_{n+1}^k , но новое направление помещается в последний столбец матрицы направлений. На $(k+1)$ -м этапе будут использоваться направления

$$[\bar{S}_1^{k+1}, \bar{S}_2^{k+1}, \dots, \bar{S}_n^{k+1}] = [\bar{S}_1^k, \bar{S}_2^k, \dots, \bar{S}_{m-1}^k, \bar{S}_{m+1}^k, \dots, \bar{S}_n^k, \bar{S}_{n+1}^k].$$

5. Критерий останова. Алгоритм прерывается, ес. изменение по каждой переменной оказывается меньше заданной точности по соответствующей переменной или $\|\bar{x}_n^k - \bar{x}_0^k\| \leq \varepsilon$.

2.Транспортная задача с ограничениями на пропускные способности: построение оптимального плана.

Транспортная задача линейного программирования с ограничениями на пропускные способности путей сообщения может быть сформулирована следующим образом. Необходимо минимизировать транспортные расходы

$$Q(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij} \rightarrow \min$$

при ограничениях

$$\left. \begin{aligned} \sum_{i=1}^m x_{ij} &= b_j, & j = \overline{1, n}, \\ \sum_{j=1}^n x_{ij} &= a_i, & i = \overline{1, m}, \\ x_{ij} &\geq 0, & i = \overline{1, m}, \quad j = \overline{1, n}, \\ x_{ij} &\leq d_{ij}, & i = \overline{1, m}, \quad j = \overline{1, n}, \end{aligned} \right\},$$

где c_{ij} - стоимость перевозки единицы продукции из пункта i в пункт j ; x_{ij} – планируемая величина перевозок из пункта i в пункт j (план перевозок X – матрица размерности $m \times n$); b_j - потребности в продукте в пункте j ; a_i - запасы в пункте i ; d_{ij} – ограничение на величину планируемой перевозки из пункта i в пункт j .

Алгоритм состоит из 2-х этапов:

- определение опорного плана;
- определение оптимального плана методом потенциалов.

Метод потенциалов для определения оптимального плана

1. Пусть у нас есть опорный план. Для перевозок вида

$$0 < x_{ij} < d_{ij}, \quad (1)$$

для определения потенциалов u_i и v_j соответствующих пунктов отправления и пунктов назначения составляется система уравнений

$$v_j - u_i = c_{ij}, \quad (2)$$

и находятся потенциалы u_i и v_j .

2. Для остальных клеток транспортной таблицы вычисляются значения

$$c_{ij} - (v_j - u_i). \quad (3)$$

$$\text{Если для } \forall x_{ij} = 0 \quad \alpha_{ij} = c_{ij} - (v_j - u_i) \geq 0, \quad (4)$$

$$\text{и для } \forall x_{ij} = d_{ij} \quad -\beta_{ij} = c_{ij} - (v_j - u_i) \leq 0, \quad (4)$$

то опорный план транспортной задачи *оптимален*.

Если эти условия не выполняются, то среди отрицательных чисел α_{ij} и β_{ij} выбираем наименьшее. Пусть это наименьшее из чисел соответствует перевозке с индексами i_0 и j_0 . Возможно 2 варианта:

а) наименьшее из чисел соответствует $x_{i_0 j_0} = 0$;

б) наименьшее из чисел соответствует $x_{i_0 j_0} = d_{i_0 j_0}$.

Начиная с перевозки $x_{i_0 j_0}$, строим замкнутый цикл из базисных перевозок транспортной задачи (соответствующих (1)). В зависимости от случаев а) и б) дальнейшие действия отличаются.

а) В этом случае для улучшения плана мы должны ввести в базис перевозку $x_{i_0 j_0}$. Пометим перевозки цикла, начиная с i_0, j_0 , поочередно знаками $+$ и $-$. ($x_{i_0 j_0}$ помечаем знаком « $+$ »).

Определим для отрицательной полуцепи $\theta' = \min \{x_{ij}^-\}$,

для положительной полу цепи $\theta'' = \min \{d_{ij}^+ - x_{ij}^+\}$

и возьмем $\theta = \min \{\theta', \theta'', d_{i_0, j_0}\}$.

Для получения более экономичного плана перевозки положительной полуцепи увеличиваем на θ , а отрицательной – уменьшаем на θ .

После этого переходим к п.1.

б) В этом случае помечаем перевозку $x_{i_0 j_0}$ помечаем знаком «-», а остальные перевозки цикла помечаем последовательно + и - .

Определим для отрицательной полу цепи $\theta' = \min \{x_{ij}^-\}$,

для положительной полу цепи $\theta'' = \min \{d_{ij}^+ - x_{ij}^+\}$.

Определяем $\theta = \min \{\theta', \theta'', d_{i_0, j_0}\}$.

Увеличиваем перевозки положительной полуцепи на θ , а перевозки отрицательной полу цепи уменьшаем на θ . Переходим к п.1.

7 билет

1.Методы первого порядка.Алгоритм наискор. спуска.

К методам первого порядка относятся алгоритмы, в которых в процессе поиска кроме информации о самой ф-ции используется информация о производных первого порядка. К группе таких методов относятся различные градиентные методы.

Алгоритм наискорейшего спуска

Градиент ф-ции в любой точке показывает направление наибольшего локального увеличения $f(\bar{x})$. Поэтому при поиске минимума $f(\bar{x})$, следует двигаться в направлении противоположном направлению градиента $\nabla f(\bar{x})$ в данной точке, то есть в направлении *наискор-го спуска*.

Итерационная ф-ла процесса наискорейшего спуска имеет вид

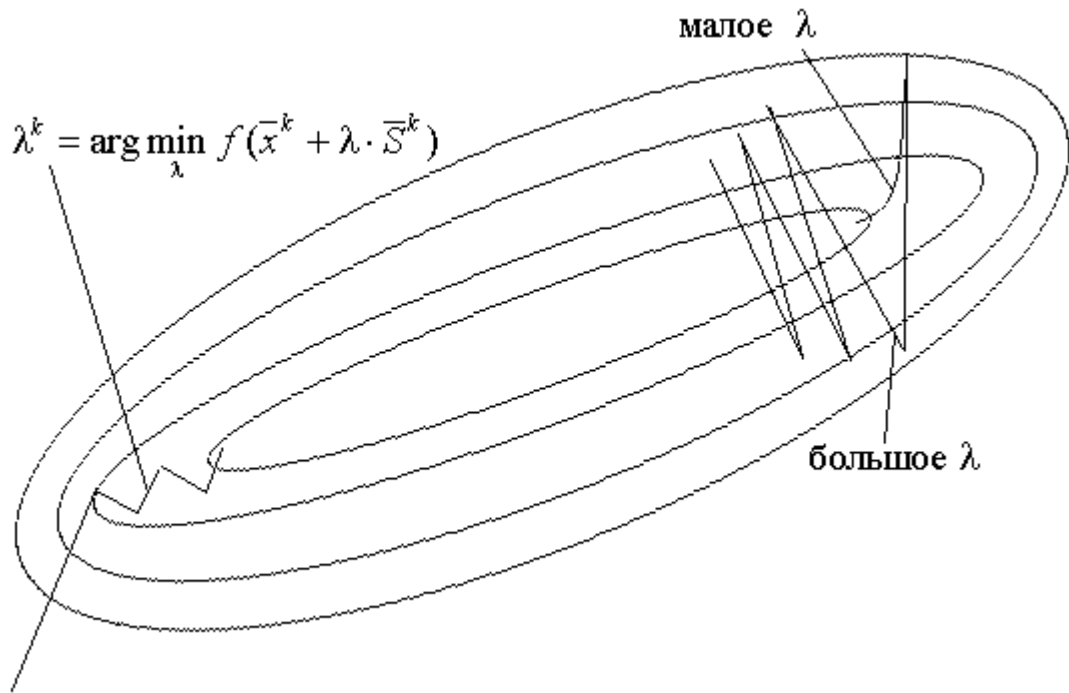
$$\bar{x}^{k+1} = \bar{x}^k - \lambda^k \cdot \nabla f(\bar{x}^k),$$

или
$$\bar{x}^{k+1} = \bar{x}^k - \lambda^k \cdot \frac{\nabla f(\bar{x}^k)}{\|\nabla f(\bar{x}^k)\|} = \bar{x}^k - \lambda^k \cdot \bar{S}^k.$$

Очевидно, что в зависимости от выбора параметра λ траектории спуска будут существенно различаться. При большом значении λ траектория спуска будет представлять собой колебательный процесс, а при слишком больших λ процесс может расходиться. При выборе малых λ траектория спуска будет плавной, но и процесс будет сходиться очень медленно.

Обычно λ выбирают из условия $\lambda^k = \arg \min_{\lambda} f(\bar{x}^k + \lambda \cdot \bar{S}^k)$, решая

одномерную задачу минимизации с использованием некоторого одномерного метода. В этом случае получаем алгоритм наискорейшего спуска.



Ес. λ определяется в результате одномерной минимизации, то градиент в точке очередного приближения будет ортогонален направлению предыдущего спуска $\nabla f(\bar{x}^{k+1}) \perp \bar{S}^k$. Вообще говоря, процедура наискорейшего спуска может закончиться в стационарной точке любого типа, в которой $\nabla f(\bar{x}) = \bar{0}$. Поэтому следует проверять, не завершился ли алгоритм в седловой точке. Эффективность алгоритма зависит от вида минимизируемой ф-ции. Алгоритм наискорейшего спуска сойдется за одну итерацию при любом начальном приближении для ф-ции $f(\bar{x}) = x_1^2 + x_2^2$, но сходимость будет очень медленной, например, в случае ф-ции вида $f(\bar{x}) = x_1^2 + 100x_2^2$. В тех ситуациях, когда линии уровня минимизируемой ф-ции представляет собой прямолинейный или, хуже того, криволинейный "овраг" эффективность алгоритма оказывается очень низкой. Очевидно, что хорошие результаты может давать предварительное масштабирование функций. Процесс наискорейшего спуска обычно быстро сходится вдали от точки экстремума и медленно в районе экстремума. Поэтому метод наискор. спуска нередко используют в комбинации с др. алгоритмами.

2. Транспортная задача с ограничениями на пропускные способности: построение опорного плана.

Построение опорного плана

Состоит из 2-х этапов. Предварительного этапа, напоминающего метод минимального элемента, и ряд этапов метода потенциалов, применяемого к расширенной задаче.

Предварительный этап разбивается на несколько однотипных шагов.

Первый шаг. Среди элементов c_{ij} матрицы C находим минимальный. Если этим элементом является $c_{i_1 j_1}$, то находим $x_{i_1 j_1} = \min \{a_{i_1}, b_{j_1}, d_{i_1 j_1}\}$. Возможны 3 случая: $x_{i_1 j_1} = a_{i_1}$; $x_{i_1 j_1} = b_{j_1}$; $x_{i_1 j_1} = d_{i_1 j_1}$.

В первом случае все остальные перевозки строки $x_{i_1 j} = 0 \quad \forall j \neq j_1$, во втором – столбца $x_{ij_1} = 0 \quad \forall i \neq i_1$. В третьем случае заполняется только $x_{i_1 j_1}$. Далее вычеркиваем из матрицы C либо строку, либо столбец, либо элемент $c_{i_1 j_1}$. Преобразуем величины в таблице

$$a_i^1 = \begin{cases} a_i, & i \neq i_1, \\ a_{i_1} - x_{i_1 j_1}, & i = i_1; \end{cases} \quad b_j^1 = \begin{cases} b_j, & j \neq j_1, \\ b_{j_1} - x_{i_1 j_1}, & j = j_1. \end{cases}$$

Второй шаг состоит в проведении тех же операций применительно к невычеркнутым элементам матрицы C , не заполненным позициям матрицы X и величинам a_i^1, b_j^1 .

Шаги предварительного этапа следуют до полного заполнения матрицы X . Согласно процессу формирования матрицы X ее элементы удовлетворяют условиям

$$\left. \begin{aligned} \sum_{i=1}^m x_{ij} &\leq b_j, & j = \overline{1, n}, \\ \sum_{j=1}^n x_{ij} &\leq a_i, & i = \overline{1, m}, \\ 0 \leq x_{ij} &\leq d_{ij}, & i = \overline{1, m}, \quad j = \overline{1, n}. \end{aligned} \right\}$$

Положим

$$\left. \begin{aligned} x_{m+1, j} &= b_j - \sum_{i=1}^m x_{ij}, & j = \overline{1, n}, \\ x_{i, n+1} &= a_i - \sum_{j=1}^n x_{ij}, & i = \overline{1, m}, \\ \varepsilon &= \sum_{i=1}^m x_{i, n+1} = \sum_{j=1}^n x_{m+1, j}. \end{aligned} \right\}$$

Если $\varepsilon = 0$, то матрица X очевидно является (опорным) планом задачи T_d . Однако в общем случае $\varepsilon > 0$, и для получения искомого опорного плана задачи T_d необходимо провести еще несколько итераций методом

потенциалов. Введем расширенную задачу $T_d(M)$, которую образуем из T_d следующим образом. Присоединим к пунктам производства задачи T_d фиктивный пункт A_{m+1} с объемом производства $a_{m+1} = \varepsilon$, а к пунктам потребления пункт B_{n+1} с $b_{n+1} = \varepsilon$. Пусть стоимость перевозки $c_{i,n+1}$, $i = \overline{1, m}$ и $c_{m+1,j}$, $j = \overline{1, n}$ равна M (максимально большое число), а $c_{m+1,n+1} = 0$. Для этой задачи легко образовать опорный план, введя перевозки из i в $n+1$, и из $m+1$ в j , равные $x_{i,n+1}$ и $x_{m+1,j}$, и взяв $x_{m+1,n+1} = 0$. К задаче $T_d(M)$ применяем метод потенциалов. При решении возможно 2 случая.

1. После ряда итераций строится опорный план X_1 задачи $T_d(M)$, согласно которому перевозка между A_{m+1} и B_{n+1} равна ε .

В этом случае множество перевозок между пунктами a_i и b_j , $i = \overline{1, m}$, $j = \overline{1, n}$ составят опорный план исходной задачи.

2. В оптимальном плане задачи $T_d(M)$, который определяется за несколько итераций метода потенциалов, перевозка между пунктами A_{m+1} и B_{n+1} меньше ε .

В этом случае задача T_d не имеет ни одного плана, т.е. неразрешима.

8 билет

1. Алгоритмы сопряженных градиентов.

МСГ: В алгоритме наискор. спуска на кажд. этапе поиска исп. только текущая информация о ф-ции $f(\bar{x}^k)$ и градиенте $\nabla f(\bar{x}^k)$. В алг-мах сопр-ых град-тов используется информация о поиске на предыдущих этапах спуска.

Направл. поиска на текущем шаге \bar{S}^k строится как линэкомбин. наискор. спуска на данном шаге $-\nabla f(\bar{x}^k)$ и направлений спуска на предыд. шагах $\bar{S}^0, \bar{S}^1, \dots, \bar{S}^{k-1}$. Веса в лин. комбин. выбираются т.о., чт. сделать эти направ-я сопряженными. В эт. случ. квадратичная ф-ция будет минимизироваться за n шагов алгоритма. При выборе весов используется только текущий градиент и градиент в предыдущей точке. В начальной точке \bar{x}^0 направление спуска $\bar{S}^0 = -\nabla f(\bar{x}^0)$: $\bar{x}^1 = \bar{x}^0 + \lambda^0 \cdot \bar{S}^0$, (1)

где λ^0 выбирается из соображений минимальности целевой ф-ции в данном направлении $\lambda^0 = \arg \min_{\lambda} f(\bar{x}^0 + \lambda \cdot \bar{S}^0)$. Новое направление

поиска $\bar{S}^1 = -\nabla f(\bar{x}^1) + \omega_1 \cdot \bar{S}^0$, (2)

где ω_1 выбирается так, чт. сделать направления \bar{S}^1 и \bar{S}^0 сопряженными по отношению к матрице H : $(\bar{S}^0)^T \cdot H \cdot \bar{S}^1 = 0$. (3)

Для квадратичной ф-ции справедливы соотношения:

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

где $\Delta \bar{x} = \bar{x} - \bar{x}^0$, $\nabla f(\bar{x}) = \nabla f(\bar{x}^0) + \nabla^2 f(\bar{x}^0) \cdot \Delta \bar{x}$.

Ес. положить $\bar{x} = \bar{x}^1$, тогда $\bar{x}^1 - \bar{x}^0 = \lambda^0 \bar{S}^0$ и

$$\nabla f(\bar{x}^1) - \nabla f(\bar{x}^0) = \nabla^2 f(\bar{x}^0) \cdot (\bar{x}^1 - \bar{x}^0) = H(\lambda^0 \bar{S}^0). \quad (4)$$

Воспользуемся (4), чт. исключить $(\bar{S}^0)^T$ из уравнения (3). Для квадратичной ф-ции $H = H^T$, поэтому транспонировав (4) и умножив справа на H^{-1} , получим $[\nabla f(\bar{x}^1) - \nabla f(\bar{x}^0)]^T H^{-1} = (\bar{x}^1 - \bar{x}^0)^T H^T H^{-1}$ и далее

$$(\bar{S}^0)^T = \frac{(\bar{x}^1 - \bar{x}^0)^T}{\lambda^0} = \frac{[\nabla f(\bar{x}^1) - \nabla f(\bar{x}^0)]^T \cdot H^{-1}}{\lambda^0} \Rightarrow, \text{ для сопряженности } \bar{S}^0 \text{ и } \bar{S}^1$$

$$[\nabla f(\bar{x}^1) - \nabla f(\bar{x}^0)]^T \cdot H^{-1} H \bar{S}^1 = [\nabla f(\bar{x}^1) - \nabla f(\bar{x}^0)]^T [-\nabla f(\bar{x}^1) + \omega_1 \bar{S}^0] = 0.$$

Вследствие изложенных ранее свойств сопряженности все перекрестные члены исчезают. Учитывая, что $\bar{S}^0 = -\nabla f(\bar{x}^0)$ и, \Rightarrow ,

$$-\nabla^T f(\bar{x}^1) \nabla f(\bar{x}^1) + \omega_1 \nabla^T f(\bar{x}^0) \nabla f(\bar{x}^0) = 0$$

$$\text{получим для } \omega_1 \text{ следующее: } \omega_1 = \frac{\nabla^T f(\bar{x}^1) \cdot \nabla f(\bar{x}^1)}{\nabla^T f(\bar{x}^0) \cdot \nabla f(\bar{x}^0)} = \frac{\|\nabla f(\bar{x}^1)\|^2}{\|\nabla f(\bar{x}^0)\|^2}. \quad (5)$$

Направл.поиска \bar{S}^2 строится в виде лин.комбин.в-ров $-\nabla f(\bar{x}^2)$, \bar{S}^0 , \bar{S}^1 , причем так,чт.оно было сопряж-м с \bar{S}^1 . Ес. распространить выкладки на $\bar{S}^2, \bar{S}^3, \dots$, опуская их содержание и учитывая, что $(\bar{S}^k)^T \nabla f(\bar{x}^{k+1}) = 0$ приводит к $\nabla^T f(\bar{x}^k) \nabla f(\bar{x}^{k+1}) = 0$, можно

$$\text{получить общее выражение для } \omega_k: \quad \omega_k = \frac{\|\nabla f(\bar{x}^k)\|^2}{\|\nabla f(\bar{x}^{k-1})\|^2}. \quad (6)$$

Все весовые коэффициенты, предшествующие ω_k , что особенно интересно, оказываются нулевыми.

Полностью алгоритм описывается следующей по=>стью действий:

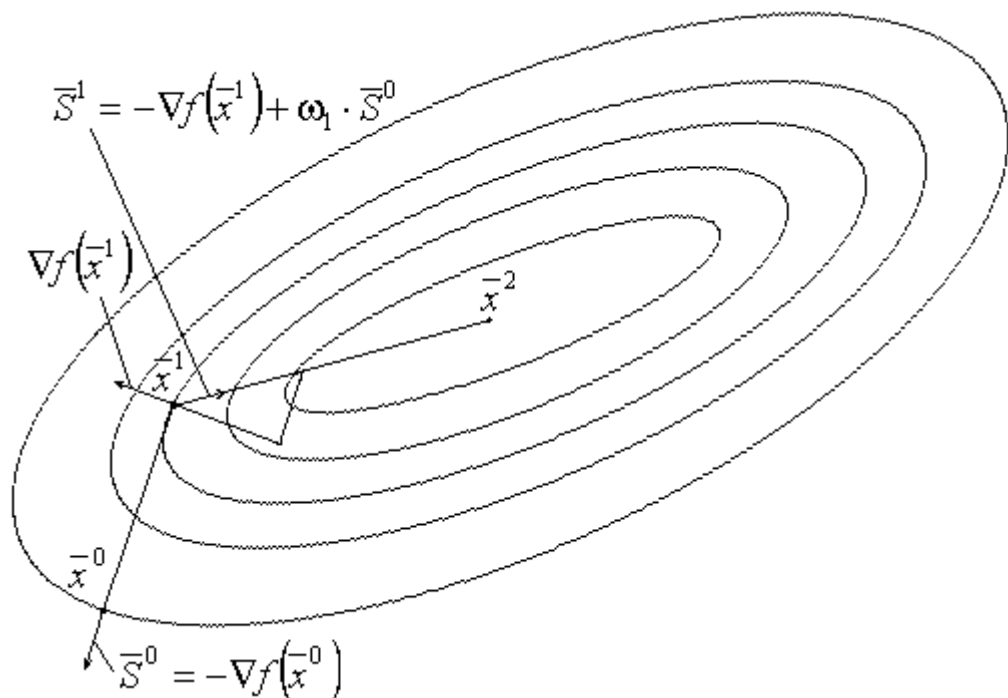
1. В точке начального приближения \bar{x}^0 вычисляется $\bar{S}^0 = -\nabla f(\bar{x}^0)$.
2. На k -м шаге с помощью одномерного поиска в направлении \bar{S}^k определяется минимум ф-ции, то есть решается задача

$$\lambda^k = \arg \min_{\lambda} f(\bar{x}^k + \lambda \cdot \bar{S}^k)$$

и находится очередное приближение $\bar{x}^{k+1} = \bar{x}^k + \lambda^k \cdot \bar{S}^k$.

3. Вычисляется $f(\bar{x}^{k+1})$ и $\nabla f(\bar{x}^{k+1})$.
4. Определяется направление $\bar{S}^{k+1} = -\nabla f(\bar{x}^{k+1}) + \omega_{k+1} \cdot \bar{S}^k$.
5. Алгоритм заканчивается, ес. $\|\bar{S}^k\| < \varepsilon$, где ε – заданная величина.

После $n+1$ итераций ($k=n$), ес. не произошел останов алгоритма, процедура циклически повторяется с заменой \bar{x}^0 на \bar{x}^{n+1} и возвратом на первый пункт алгоритма. Ес. исходная ф-ция явл-ся квадратичной, то $(n+1)$ -е приближение даст точку экстремума данной ф-ции. Описанный алгоритм с построением ω_k по ф-лам (6) соответствует методу сопряженных градиентов **Флетчера-Ривса**.



В модификации **Полака-Рибьера** (Пшеничного) метод сопряженных градиентов отличается только вычислением

$$\omega_k = \frac{\nabla^T f(\bar{x}^k) \cdot [\nabla f(\bar{x}^k) - \nabla f(\bar{x}^{k-1})]}{\nabla^T f(\bar{x}^{k-1}) \cdot \bar{S}^{k-1}}. \quad (7)$$

В случае квадратичных функций обе модификации примерно эквивалентны, однако в модификации Полака-Рибьера алгоритм иногда оказывается несколько эффективнее.

2. Транспортная задача по критерию времени.

В такой транспортной задаче решающую роль играет не стоимость перевозок, а время, которое затрачивается на доставку груза. Оптимальным планом считается план, который минимизирует время перевозок. Подобные задачи возникают при перевозках скоропортящихся продуктов и в военном деле, где зачастую стоимость перевозок играет второстепенную роль. Как и в предыдущей задаче имеется m пунктов отправления, с запасами однородного продукта a_i , n пунктов назначения с потребностями b_j .

Задача закрытого типа, т.е.
$$\sum_{j=1}^n b_j = \sum_{i=1}^m a_i.$$

Задана матрица $T = [t_{ij}]_{m \times n}$, где t_{ij} – необходимое для перевозки груза из пункта i в пункт j . Необходимо выбрать среди допустимых такой план $X = [x_{ij}]_{m,n}$, что

$$\left. \begin{aligned} \sum_{i=1}^m x_{ij} &= b_j, \forall j \in \overline{1, n} \\ \sum_{j=1}^n x_{ij} &= a_i, \forall i \in \overline{1, m} \end{aligned} \right\}$$

и грузы будут доставляться по этому плану за минимальное время T_{\min} .

Каждому допустимому плану $X = [x_{ij}]_{m,n}$ соответствует некоторый набор $\{t_{ij}\}_X$, состоящий из элементов матрицы $T = [t_{ij}]_{m \times n}$, соответствующих положительным компонентам x_{ij} плана X . Т.е. t_{ij} включается в набор, если производится перевозка из пункта i в пункт j .

Время t_X , необходимое для выполнения плана X , определяется следующим образом
$$t_X = \max\{t_{ij}\}_X.$$

Тогда время, необходимое для реализации оптимального плана X^*

$$t_{X^*} = \min_X t_X = \min_X (\max\{t_{ij}\}_X).$$

Алгоритм отыскания оптимального решения

Данный алгоритм состоит из двух этапов:

1. Предварительный шаг.

Строим допустимый план по методу северо-западного угла или минимального элемента X_0 .

2. Общий шаг.

Просматриваем все t_{ij} , соответствующие положительным x_{ij} и выбираем

из них наибольшее $t'_{ij} = \max_{x_{ij}>0} \{t_{ij}\}$ и вычеркиваем все клетки, для

которых $t_{ij} \geq t'_{ij}$, $\forall x_{ij} = 0$. Далее производим исправление плана X_0 , для чего стремимся обратить в 0 перевозку x'_{ij} , соответствующую

$t'_{ij} = \max_{x_{ij}>0} \{t_{ij}\}$ (в той же клетке). Если это удастся, то, естественно,

уменьшается время, необходимое на реализацию нового допустимого плана X_1 . Для построения плана X_1 строится цикл как и в методе потенциалов.

Первой клеткой отрицательной полуцепи берем клетку с t'_{ij} , остальными клетками отрицательной полуцепи выбираем клетки с $x_{ij}>0$, клетками положительной полуцепи берем клетки с $t_{ij} < t'_{ij}$.

Затем перемещаем минимальный элемент θ отрицательной полуцепи в положительную. Если удастся обратить x'_{ij} в 0, то реализация нового плана потребует меньшего времени.

Общий шаг продолжаем повторять до тех пор, пока станет невозможным обращение в 0 всей перевозки x'_{ij} из клетки с максимальным временем t'_{ij} .

9 билет

1. Многопараметрический поиск.

Милем и Кентреллом предложен метод поиска, основанный на использовании двух подбираемых параметров для минимизации $f(\bar{x})$ в каждом направлении поиска. В этом алгоритме по-прежнему действия определяются ф-лой: $\bar{x}^{k+1} = \bar{x}^k - \lambda_0^k \cdot \nabla f(\bar{x}^k) + \lambda_1^k \cdot \Delta \bar{x}^{k-1}$, (1)

где $\Delta \bar{x}^{k-1} = \bar{x}^k - \bar{x}^{k-1}$.

На каждом шаге решается задача минимизации по двум параметрам:

$$\min_{\lambda_0, \lambda_1} f(\bar{x}^k - \lambda_0 \cdot \nabla f(\bar{x}^k) + \lambda_1 \cdot \Delta \bar{x}^{k-1}).$$

После чего находится очередное приближение по ф-ле (1). При этом можно показать, что

$$\nabla^T f(\bar{x}^k) \nabla f(\bar{x}^{k+1}) = 0, \quad \nabla^T f(\bar{x}^{k+1}) \Delta \bar{x}^{k+1} = 0 \quad \text{и} \quad \nabla^T f(\bar{x}^{k+1}) \Delta \bar{x}^k = 0.$$

На первом шаге $\Delta \bar{x}^{-k-1} = 0$, а \bar{x}^0 должно быть задано. На k -м шаге:

1. Вычисляется \bar{x}^k , $\nabla f(\bar{x}^k)$ и $\Delta \bar{x}^{-k-1} = \bar{x}^k - \bar{x}^{k-1}$.
2. Пользуясь одним из эффективных методов, например, методом Ньютона находятся с требуемой точностью λ_0^k и λ_1^k .
3. По соотношению (1) вычисляют \bar{x}^{k+1} и переходят к пункту 1.
4. Каждый $(n+1)$ -й шаг начинается с $\Delta \bar{x}^{-k-1} = 0$.
5. Процесс заканчивается, когда $|\Delta f(\bar{x})| < \varepsilon$.

На квадратичных ф-циях алгоритм по эффективности близок к методу сопряженных градиентов.

Крэгг и Леви распространили данный метод на случай большего числа параметров. На каждом шаге очередное приближение находится как

$$\bar{x}^{k+1} = \bar{x}^k - \lambda_0 \cdot \nabla f(\bar{x}^k) + \sum_{i=1}^m \lambda_i \cdot \Delta \bar{x}^{-i-1}$$

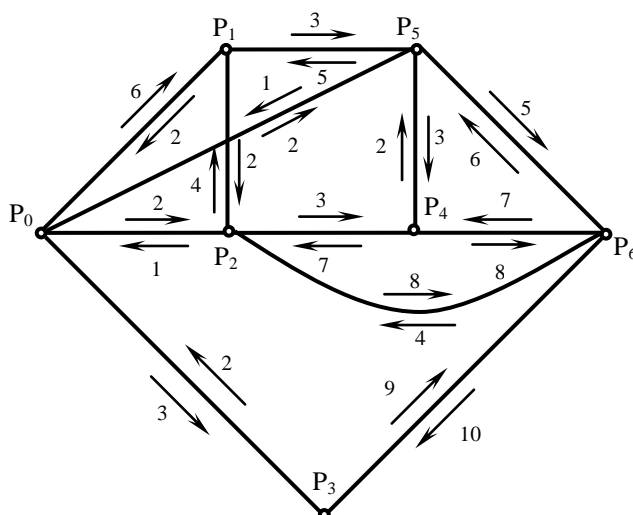
при $m \leq n-1$, а, \Rightarrow , на каждом шаге при минимизации $f(\bar{x})$ в заданном направлении решается задача вида

$$\min_{\lambda_0, \lambda_1, \dots, \lambda_m} f\left(\bar{x}^k - \lambda_0 \cdot \nabla f(\bar{x}^k) + \sum_{i=1}^m \lambda_i \cdot \Delta \bar{x}^{-i-1}\right).$$

Достоинства и недостатки такого подхода очевидны.

2. Задача о максимальном потоке в транспортной сети.

Определение 1. Транспортной сетью называется конечный граф G , состоящий из $(n+1)$ вершин P_0, P_1, \dots, P_n и из дуг (P_i, P_j) , соединяющих некоторые пары этих вершин, причем каждой дуге поставлено в соответствие число $c_{ij} \geq 0$, называемое пропускной способностью дуги (P_i, P_j) . Вершина



P_0 называется входом сети, а P_n – выходом транспортной сети. Будем считать, что граф симметрический, т.е. если в него входит дуга (P_i, P_j) , то входит и дуга (P_j, P_i) .

c_{ij} – определяет количество вещества (машин, и т.п.), которое может протекать по дуге в единицу времени.

Например:

$c_{ij} = 0$, если P_i и P_j не

соединены дугой. По путям $\mu(P_0, P_{i_1}, P_{i_2}, \dots, P_{i_k}, P_n)$, составленным из дуг $(P_0, P_{i_1}), (P_{i_1}, P_{i_2}), \dots, (P_{i_k}, P_n)$ сети направляется транспорт из P_0 в P_n .

Потоком x_{ij} по дуге (P_i, P_j) ($i, j = 0, \dots, n, i \neq j$) наз-ся кол-во вещества, проходящее через эту дугу в единицу времени. Потоком по сети или просто потоком будем называть совокупность $\{x_{ij}\}$ потоков по всем дугам сети. Потоки

должны удовлетворять следующим ограничениям:

1. $0 \leq x_{ij} \leq c_{ij}$ ($i, j = 0, \dots, n, i \neq j$),
2. $\sum_{i=0}^{n-1} x_{ik} = \sum_{j=1}^n x_{kj}$ ($k = 1, \dots, n-1$), что означает: количество вещества, притекающее в вершину сети равно количеству вещества, вытекающего из него (кроме P_0 и P_n).

Поток, удовлетворяющий ограничениям 1 и 2 будем называть допустимым.

Из (2) видно, что общее количество вещества, вытекающего из P_0 , $\sum_{j=1}^n x_{0j}$,

совпадает с общим количеством вещества, притекающего в P_n , $\sum_{i=0}^{n-1} x_{in}$, т.е.

3. $\sum_{i=0}^{n-1} x_{in} = \sum_{j=1}^n x_{0j} = Q$. Линейная форма Q называется потоком по сети.

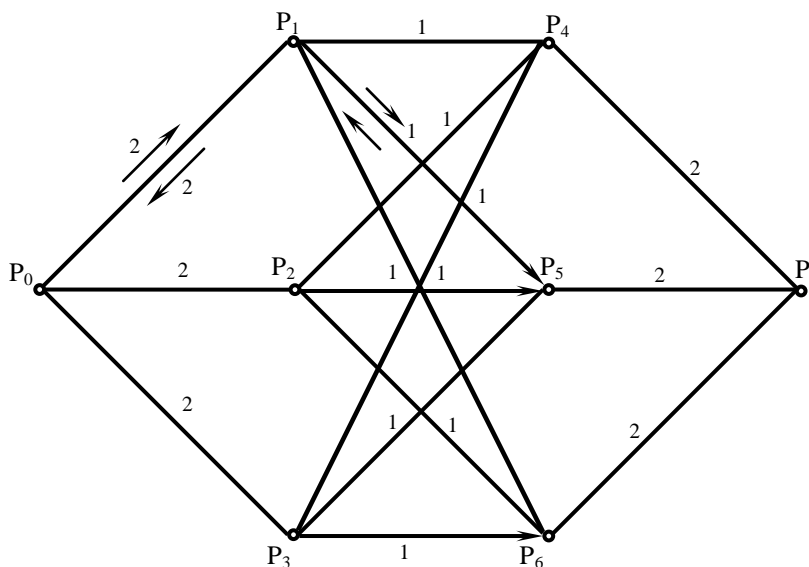
Задача о максимальном потоке в транспортной сети заключается в отыскании такого решения x_{ij}^* ($i, j = \overline{0, n}$) системы (1 – 2) (т.е. такого

допустимого потока), который максимизирует Q .

Это решение $\{x_{ij}^*\}$ называется

максимальным потоком сети.

Рассмотрим такой простейший пример:



Найти максимальный поток из P_0 в P_7 . Толстые дуги насы-

щены, т.е. $x_{ij} = c_{ij}$, полный поток $\sum_{j=1}^n x_{0j} = 5$, а максимальный поток равен

6. Разобьем множество всех вершин G на два подмножества U и V так, что $P_0 \in U$ и $P_n \in V$. Сечением (U, V) сети G назовем совокупность всех дуг (P_i, P_j) , концы которых принадлежат разным подмножествам.

Каждому сечению поставим в соответствие неотрицательное число $C(U, V)$ – пропускную способность сечения, равную сумме c_{ij} всех дуг сечения, начинающихся в U и кончающихся в V , т.е. $C(U, V) = \sum_{\substack{P_i \in U \\ P_j \in V}} c_{ij}$

Любой путь из P_0 в P_n обязательно содержит хотя одну дугу сечения (U, V) , которая начинается в U и заканчивается в V . Ясно, что пропускная способность пути не превышает пропускной способности каждой его дуги. Поэтому величина любого потока из P_0 в P_n , которая является суммарной величиной пропускной способности всех путей из P_0 в P_n , не может превысить пропускной способности любого сечения (U, V) , т.е. всегда $Q \leq C(U, V)$.

Теорема Форда-Фалкерсона утверждает: Для заданной транспортной сети наибольшая величина потока равна наименьшей пропускной способности сечения, т.е. $\max Q \leq \min_{(U, V)} C(U, V) = C(U^*, V^*)$

по всем возможным сечениям. Т.е. если удастся построить такой поток $\{x_{ij}^*\}$, что $Q^* = C(U^*, V^*)$, то этот поток будет максимальным, а сечение $C(U^*, V^*)$ – обладать минимальной пропускной способностью.

Естественно, что каждую транспортную сеть стремятся использовать оптимально, т.е. организовать перевозки по транспортным путям таким образом, чт. поток перевозимых грузов был максимальным.

Алгоритм построения максимального потока в транспортной сети

Предварительный шаг:

Условия 1. записываются в виде следующей таблицы:

Если $c_{ij} > 0$, $c_{ji} = 0$, то в клетке ji ставим 0,

Если же $c_{ij} = c_{ji} = 0$, то клетки ij и ji не заполняем.

Начальная таблица: (4)

	P_0	...	P_i	...	P_j	...	P_n
P_0			c_{0i}		c_{0j}		c_{0n}
...							

P_i	$c_{i0} = 0$				c_{ij}		c_{in}
...							
P_j	$c_{j0} = 0$		c_{ji}				c_{jn}
...							
P_n	$c_{n0} = 0$		$c_{ni} = 0$		$c_{nj} = 0$		

Общий шаг: (из трех действий)

1. Отыскание по таблице нового пути из P_0 в P_n . Сначала отмечаем P_0 -й столбец *. Затем отыскиваем в строке P_0 все положительные c_{0i} и содержащие их столбцы отмечаем сверху числом 0 (номером вершины P_0). Т.е. выделили все дуги (P_0, P_i) , которые могут быть первыми дугами различных путей из P_0 в P_n . Просматриваем затем строки затем строки, имеющие те же номера, что и отмеченные столбцы. В каждой такой строке (например P_i) отыскиваем все неотрицательные c_{ij} , расположенные в неотмеченных столбцах, и отмечаем эти столбцы номером рассматриваемой строки (например i).

Этим самым будут выделены дуги (P_i, P_j) с положительной пропускной способностью, которые могут служить вторыми дугами различных путей из P_0 в P_n , т.е. уже выделены (P_0, P_i) и (P_i, P_j) .

Продолжаем аналогичный просмотр строк с номерами отмеченных столбцов. Процесс оканчивается, если:

а) Отмечен P_n -й столбец, т.е. удалось выделить дугу (P_k, P_n) с $c_{kn} > 0$, которая служит последней дугой некоторого пути из P_0 в P_n .

б) Просмотрены все строки и нельзя отметить новых столбцов (т.е. в неотмеченных столбцах нет $c_{ij} > 0$), это означает отсутствие пути из P_0 в P_n , все дуги которого обладают положительными пропускными способностями (Алгоритм закончен). В случае а) искомый новый путь из P_0 в P_n отыскиваем следующим образом. Начиная от P_n . Пусть столбец P_n был отмечен номером k , т.е. предшествующая вершина в пути, соединяющем P_0 с P_n , – P_k . (При просмотре строки P_k был отмечен столбец P_n). Число $c_{kn} > 0$ помечаем знаком “–” (c_{kn}^-). Число c_{nk} , расположенное симметрично к диагонали, помечаем знаком “+” (c_{nk}^+). Раз рассматривалась P_k -я строка, значит перед этим был отмечен P_k столбец номером l (например). По столбцу P_k движемся вверх до P_l -ой строки. c_{lk} отмечаем “–” (c_{lk}^-), а c_{kl} знаком “+”. Этот процесс продолжаем до тех пор, пока не придем к P_0 -ой строке и не отметим элемент этой строки и симметричный элемент.

2. Определяем пропускную способность найденного пути $\theta = \min\{c_{ij}^-\}$

3. Вычисляем новые пропускные способности дуг найденного пути и симметричных с ними $\forall c_{ij}^-, c_{ij} = c_{ij}^- - \theta$,

$$\forall c_{ij}^+, c_{ij} = c_{ij}^+ + \theta.$$

В результате получаем новую таблицу, с которой повторяем эти 3 шага.

θ представляет собой пропускную способность найденного пути. Если дуга (P_i, P_j) входила в предыдущий путь, то по ней уже пропущено θ вещества. Если (P_i, P_j) входит в один путь, то естественно, что в этом пути по ней нельзя пропустить больше чем $c_{ij} - \theta$ вещества в единицу времени.

После действия 3 получаем таблицу для сети G с новыми пропускными способностями. Все старые отметки убираются и возвращаемся к действию 1 общего шага, который применяем до тех пор, пока не придем к окончательной таблице, в которой нет ни одного пути из P_0 в P_n .

По этой таблице легко определить любую дугу, по которой протекает поток. Пропускная способность дуг, по которым протекает поток, уменьшилась по сравнению с c_{ij} на величину x_{ij} , а пропускная способность противоположно направленной (симметричной) дуги увеличилась на x_{ij} , что свидетельствует об отсутствии потока по ней.

Обозначим через U множество вершин сети G , которые достижимы из P_0 по некоторому пути в последней таблице, а множество остальных вершин через V .

Тогда увидим, что все дуги сечения (U, V) , направленные из U в V , загружены полностью до пропускных способностей (т.е. x_{ij} для (P_i, P_j) , что $P_i \in U, P_j \in V, x_{ij} = c_{ij}$), а дуги (P_j, P_i) противоположного направления, идущие из V в U в построенном потоке не используются (т.е. $x_{ij} = 0, P_i \in U, P_j \in V$), так что величина потока равна:

$$Q = \sum_{\substack{P_i \in U \\ P_j \in V}} x_{ij} - \sum_{\substack{P_i \in U \\ P_j \in V}} x_{ji} = \sum_{\substack{P_i \in U \\ P_j \in V}} c_{ij} = C(U, V)$$

т.е. построенный поток максимален, а (U, V) – сечение с минимальной пропускной способностью.

Для определения полученного максимального потока $\{x_{ij}^*\}$, $i, j = \overline{0, n}$, вычитаем из всех элементов начальной таблицы (4) соответствующие элементы таблицы, полученной на последнем шаге. Положительные значения найденных разностей дают величины потоков x_{ij}^* по дугам (P_i, P_j) , а величины потока в сети вычисляются по следующей формуле:

$$Q = \sum_{i=0}^{n-1} x_{in}^* = \sum_{j=1}^n x_{0j}^*.$$

10 билет

1.Метод Ньютона и его модификации.

Методы второго порядка. В методах второго порядка при поиске минимума используют информацию о ф-ции и ее производных до второго порядка включительно. К этой группе относят метод Ньютона и его модификации. В основе метода лежит квадратичная аппроксимация $f(\bar{x})$, которую можно получить, отбрасывая в рядах Тейлора члены третьего и более высокого порядков:

$$f(\bar{x}) \approx f(\bar{x}^k) + \nabla^T f(\bar{x}^k) \cdot (\bar{x} - \bar{x}^k) + \frac{1}{2} (\bar{x} - \bar{x}^k)^T \cdot \nabla^2 f(\bar{x}^k) \cdot (\bar{x} - \bar{x}^k), \quad (1)$$

где $\nabla^2 f(\bar{x}^k) = H(\bar{x}^k)$ – матрица Гессе, представляющая собой квадратную матрицу вторых частных производных $f(\bar{x})$ в точке \bar{x}^k .

Направление поиска \bar{S}^k в методе Ньютона определяется следующим образом. Ес. заменить в выражении (1) \bar{x} на \bar{x}^{k+1} и обозначить $\Delta \bar{x}^k = \bar{x}^{k+1} - \bar{x}^k$, то получим

$$f(\bar{x}^{k+1}) \approx f(\bar{x}^k) + \nabla^T f(\bar{x}^k) \cdot \Delta \bar{x}^k + \frac{1}{2} (\Delta \bar{x}^k)^T \cdot \nabla^2 f(\bar{x}^k) \cdot (\Delta \bar{x}^k), \quad (2)$$

Минимум ф-ции $f(\bar{x})$ в направлении $\Delta \bar{x}^k$ определяется дифференцированием $f(\bar{x})$ по каждой из компонент $\Delta \bar{x}$ и приравниванием к нулю полученных выражений $\nabla f(\bar{x}^k) + \nabla^2 f(\bar{x}^k) \cdot \Delta \bar{x}^k = \bar{0}$. (3)

Это приводит к $\Delta \bar{x}^k = -[\nabla^2 f(\bar{x}^k)]^{-1} \cdot \nabla f(\bar{x}^k),$ (4)

$$\bar{x}^{k+1} = \bar{x}^k - [\nabla^2 f(\bar{x}^k)]^{-1} \cdot \nabla f(\bar{x}^k). \quad (5)$$

В данном случае и величина шага и направление поиска точно определены. Ес. $f(\bar{x})$ - квадратичная ф-ция (выпуклая вниз), то для достижения минимума достаточно одного шага.

Но в общем случае нелинейной ф-ции $f(\bar{x})$ за один шаг минимум не достигается. Поэтому итерационную ф-лу (5) обычно приводят к виду:

$$\bar{x}^{k+1} = \bar{x}^k - \lambda^k \cdot \frac{[\nabla^2 f(\bar{x}^k)]^{-1} \cdot \nabla f(\bar{x}^k)}{\|[\nabla^2 f(\bar{x}^k)]^{-1} \cdot \nabla f(\bar{x}^k)\|}, \quad (6)$$

где λ^k - параметр длины шага, или к виду

$$\bar{x}^{k+1} = \bar{x}^k - \lambda^k \cdot [\nabla^2 f(\bar{x}^k)]^{-1} \cdot \nabla f(\bar{x}^k) = \bar{x}^k - \lambda^k \cdot H^{-1}(\bar{x}^k) \cdot \nabla f(\bar{x}^k). \quad (7)$$

Направление поиска определяется вектором $\bar{S}^k = -H^{-1}(\bar{x}^k) \cdot \nabla f(\bar{x}^k)$.

Итерационный процесс (6) или (7) продолжается до тех пор, пока не будет выполнен некоторый критерий останова.

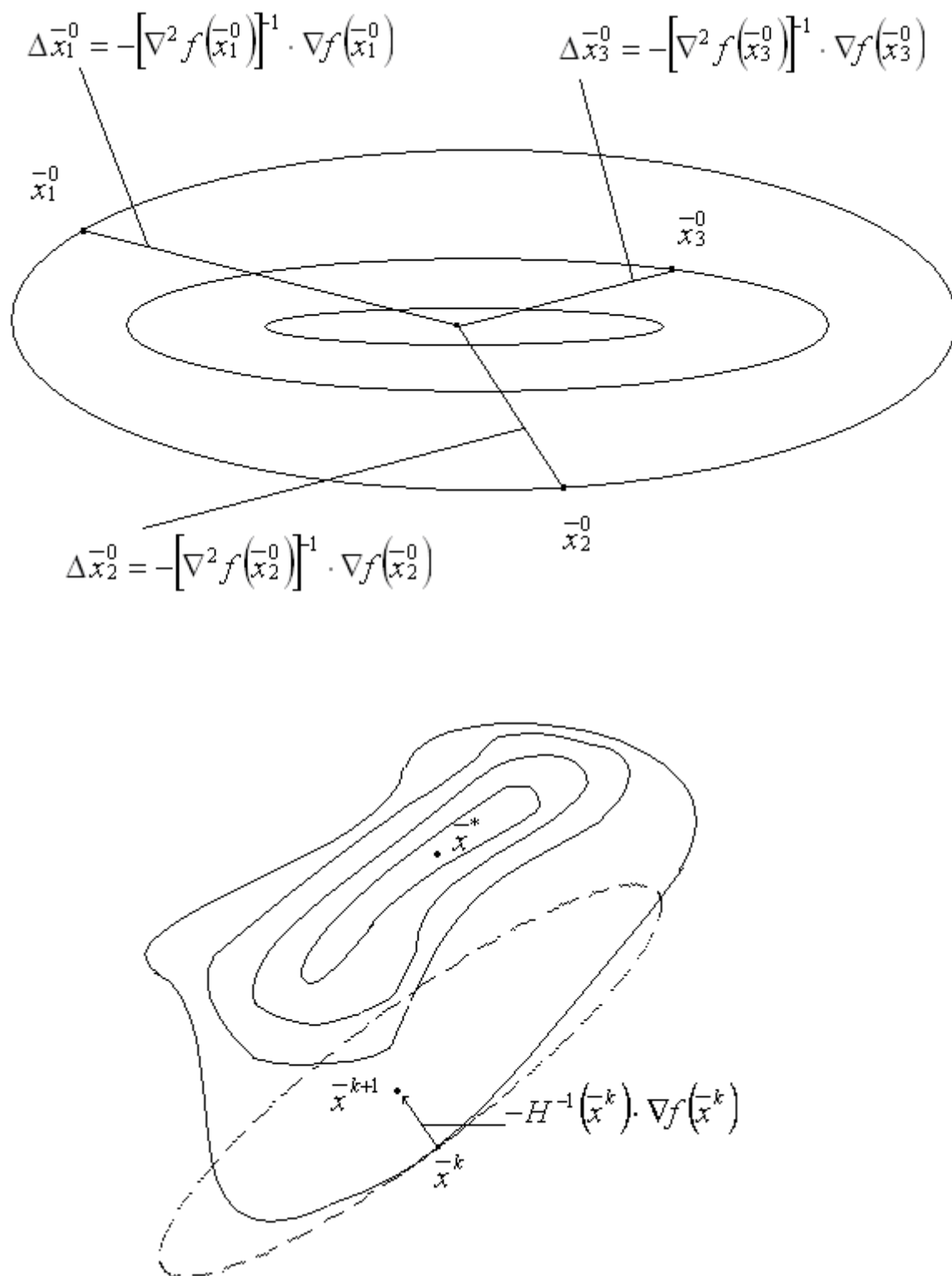


Рис.

Условием, гарантирующим сходимость метода Ньютона в предположении, что ф-ция $f(\bar{x})$ дважды дифференцируема, заключается в том, что матрица $H^{-1}(\bar{x}^k)$ должна быть положительно определенной.

Иногда определенную сложность вызывает вычисление на каждом шаге матрицы $H^{-1}(\bar{x}^k)$. Тогда вместо метода Ньютона используют его модификацию. Суть модифицированного метода Ньютона заключается в том, что при достаточно хорошем начальном приближении вычисляется матрица $[\nabla^2 f(\bar{x}^0)]^{-1}$ и в дальнейшем на всех итерациях вместо $[\nabla^2 f(\bar{x}^k)]^{-1}$ используется $[\nabla^2 f(\bar{x}^0)]^{-1}$.

Очередные приближения определяются соотношением

$$\bar{x}^{k+1} = \bar{x}^k - \lambda^k \cdot [\nabla^2 f(\bar{x}^0)]^{-1} \cdot \nabla f(\bar{x}^k) = \bar{x}^k - \lambda^k \cdot H^{-1}(\bar{x}^0) \cdot \nabla f(\bar{x}^k). \quad (8)$$

Естественно, что число итераций, необх.е для достижения минимума, обычно возрастает, но в целом процесс может оказаться экономичнее.

Градиентные методы, в частности метод наискорейшего спуска, обладают линейной скоростью сходимости. Метод Ньютона обладает квадратичной скоростью сходимости.

Применение метода Ньютона оказывается очень эффективным при условии, что выполняются необходимые и достаточные условия его сходимости. Однако само исследование необходимых и достаточных условий сходимости метода в случае конкретной $f(\bar{x})$ м.б. достаточно сложной задачей.

2.Алгоритм Пауэлла.

(см. 6-ой билет 1-ый вопрос)

11 билет

1.Методы переменной метрики.

Методы переменной метрики называют также квазиньютоновскими или градиентными с большим шагом. В этих методах в процессе поиска осуществляется аппроксимация матрицы вторых частных производных или обратной к ней. Причем для этого используются только первые производные. Очередное приближение \bar{x}^{k+1} в этих методах находится по ф-ле:

$$\bar{x}^{k+1} = \bar{x}^k + \lambda^k \cdot S^k = \bar{x}^k - \lambda^k \cdot \eta(\bar{x}^k) \cdot \nabla f(\bar{x}^k), \quad (1)$$

где матрица $\eta(\bar{x}^k)$, которую иногда называют матрицей направлений, представляет собой аппроксимацию $H^{-1}(\bar{x}^k) = [\nabla^2 f(\bar{x}^k)]^{-1}$.

Для квадратичной целевой ф-ции (или квадратичной аппроксимации целевой ф-ции) имеем:

$$f(\bar{x}) \approx f(\bar{x}^k) + \nabla^T f(\bar{x}^k) \cdot (\bar{x} - \bar{x}^k) + \frac{1}{2} (\bar{x} - \bar{x}^k)^T \cdot \nabla^2 f(\bar{x}^k) \cdot (\bar{x} - \bar{x}^k),$$

где $\nabla^2 f(\bar{x}^k) = H(\bar{x}^k)$.

Ес. вместо \bar{x} подставить в ф-лу \bar{x}^{k+1} и продифференцировать, то получим

$$\begin{aligned} \nabla f(\bar{x}^{k+1}) &= \nabla f(\bar{x}^k) + H(\bar{x}^k) \cdot (\bar{x}^{k+1} - \bar{x}^k), \\ \nabla f(\bar{x}^{k+1}) - \nabla f(\bar{x}^k) &= H(\bar{x}^k) \cdot (\bar{x}^{k+1} - \bar{x}^k). \end{aligned}$$

Умножив на $H^{-1}(\bar{x}^k)$, получаем $\bar{x}^{k+1} - \bar{x}^k = H^{-1}(\bar{x}^k) \cdot [\nabla f(\bar{x}^{k+1}) - \nabla f(\bar{x}^k)]$. (2)

При этом, ес. $f(\bar{x})$ квадратичная ф-ция, то $H(\bar{x}^k) = H$ – постоянная матрица. Уравнение (2) можно рассматривать как систему n линейных уравнений с n неизвестными параметрами, которые необх. оценить для того, чт. аппроксимировать $H^{-1}(\bar{x})$ или $H(\bar{x})$ при заданных значениях $f(\bar{x})$, $\nabla f(\bar{x})$ и $\Delta \bar{x}$ на более ранних этапах поиска.

Для решения этих линейных уравнений могут быть использованы различные методы, каждый из которых приводит к различным методам переменной метрики. В большой группе методов матрица $H^{-1}(\bar{x}^{k+1})$ аппроксимируется с помощью информации, полученной на предыдущем k -м шаге: $H^{-1}(\bar{x}^{k+1}) \approx \omega \cdot \eta^{k+1} = \omega \cdot (\eta^k + \Delta \eta^k)$, (3)

где η^k – матрица, аппроксимирующая $H^{-1}(\bar{x}^k)$ на предыдущем шаге. Вообще $\eta^k = \eta^k(\bar{x}^k)$. В (3) $\Delta \eta^k$ представляет собой определяемую матрицу, а ω – масштабный (постоянный) множитель, в большинстве случаев равный единице. Выбор $\Delta \eta^k$ по существу и определяет соответствующий метод переменной метрики.

Для обеспечения сходимости метода матрица $\omega \cdot \eta^{k+1}$ должна быть положительно определенной и удовлетворять уравнению (2) в том случае, когда она заменяет H^{-1} .

На $(k+1)$ -м шаге мы знаем \bar{x}^k , $\nabla f(\bar{x}^k)$, $\nabla f(\bar{x}^{k+1})$ и η^k и нам требуется вычислить η^{k+1} так, чт. удовлетворялось соотношение (2).

Из (2) с учетом (3) $\Delta \bar{x}^k = \omega \cdot \eta^{k+1} \cdot [\nabla f(\bar{x}^{k+1}) - \nabla f(\bar{x}^k)] = \omega \cdot \eta^{k+1} \Delta g^k$

$$\text{И} \quad \eta^{k+1} \cdot \Delta g^k = \frac{1}{\omega} \cdot \Delta \bar{x}^k. \quad (4)$$

Т.к. $\eta^{k+1} = \eta^k + \Delta \eta^k$, то на основании (4) уравнение

$$\Delta\eta^k \cdot \Delta\bar{g}^k = \frac{1}{\omega} \cdot \Delta\bar{x}^k - \eta^k \cdot \Delta\bar{g}^k \quad (5) \text{ следует разрешить относительно } \Delta\eta^k.$$

Прямой подстановкой результата можно показать, что уравнение (5) имеет следующее решение:

$$\Delta\eta^k = \frac{1}{\omega} \cdot \frac{\Delta\bar{x}^k \cdot \bar{y}^T}{\bar{y}^T \cdot \Delta\bar{g}^k} - \frac{\eta^k \cdot \Delta\bar{g}^k \cdot \bar{z}^T}{\bar{z}^T \cdot \Delta\bar{g}^k}, \quad (6)$$

где \bar{z} и \bar{y} – произвольные вектора размерности n .

Например, ес. для $\omega = 1$ выбирается специальная линейная комбинация двух направлений $\Delta\bar{x}^k$ и $\eta^k \cdot \Delta\bar{g}^k$: $\bar{y} = \bar{z} = \Delta\bar{x}^k - \eta^k \cdot \Delta\bar{g}^k$,

то получают метод **Бройдена**. Ес. же берется $\bar{y} = \Delta\bar{x}^k$, $\bar{z} = \eta^k \cdot \Delta\bar{g}^k$,

то матрицу η^{k+1} строится в соответствии с алгоритмом **Дэвидона-Флетчера-Пауэлла**.

Так \bar{z} и \bar{y} – произвольные векторы, то оказываются допустимыми и другие возможности. Ес. в этих алгоритмах шаги $\Delta\bar{x}^k$ строятся по=> в результате минимизации ф-ции $f(\bar{x})$ в направлении \bar{S}^k , то все методы, с помощью которых вычисляют симметрическую матрицу η^{k+1} , удовлетворяющую соотношению (4), дают направления, являющиеся взаимно сопряженными (в случае квадратичной целевой ф-ции).

2.Алгоритмы сопряженных градиентов.

(см. 8-ой билет 1-ый вопрос)

12 билет

1.Алгоритм Бройдена.

Бройден показал, что ес. $\Delta\eta^k$ оказывается симметрической матрицей с рангом равным единице и должно удовлетворяться соотношение

$$\eta^{k+1} \cdot \Delta\bar{g}^k = \Delta\bar{x}^k,$$

то единственным возможным выбором $\Delta\eta^k$ явл-ся соотношение:

$$\Delta\eta^k = \frac{[\Delta\bar{x}^k - \eta^k \cdot \Delta\bar{g}^k] \cdot [\Delta\bar{x}^k - \eta^k \cdot \Delta\bar{g}^k]^T}{[\Delta\bar{x}^k - \eta^k \cdot \Delta\bar{g}^k]^T \cdot \Delta\bar{g}^k}, \quad (7)$$

где $\Delta\bar{x}^k = \bar{x}^{k+1} - \bar{x}^k$, $\Delta\bar{g}^k = \nabla f(\bar{x}^{k+1}) - \nabla f(\bar{x}^k)$.

Последовательность шагов алгоритма:

1. Задается начальное приближение \bar{x}^0 и некоторая положительно определенная матрица η^0 (например, единичная $\eta^0 = E$).

2. $\bar{x}^{k+1} = \bar{x}^k - \lambda^k \cdot \eta(\bar{x}^k) \cdot \nabla f(\bar{x}^k)$, так, что $\lambda^k = \arg \min_{\lambda} f(\bar{x}^k - \lambda \cdot \eta(\bar{x}^k) \cdot \nabla f(\bar{x}^k))$.
3. Нах-ся очередн.прибл-ние м-цы $\eta^{k+1} = \eta^k + \Delta\eta^k$, где $\Delta\eta^k$ находится по(7).
4. Проверяется критерий останова, например, $\|\nabla f(\bar{x}^{k+1})\| \leq \varepsilon$. Ес. он не выполняется, то на шаг 2.

Ес. целевая ф-ция явл-ся квадратичной, то направления поиска

$\bar{S}^k = -\eta(\bar{x}^k) \cdot \nabla f(\bar{x}^k)$ на последующих итерациях оказываются

сопряженными и для определения минимума оказывается достаточным

сделать n шагов. В случае минимизации неквадратичной ф-ции возможны нежелательные явления, например:

- 1) Матрица η^{k+1} может перестать быть положительно определенной. В этом случае необх. обеспечить ее положительную определенность каким-либо способом.
- 2) Вычисляемая величина $\Delta\eta^k$ вследствие ошибок округления может стать неограниченной.
- 3) Ес. $\Delta\bar{x}^k = -\lambda^k \cdot \eta(\bar{x}^k) \cdot \nabla f(\bar{x}^k)$ на текущем шаге случайно совпадает с направлением поиска на предыдущем шаге, то матрица η^{k+1} становится сингулярной или неопределенной.

Чт. избежать этих явлений, стараются обновлять алгоритм после n шагов, считая $(n + 1)$ -ю итерацию начальной.

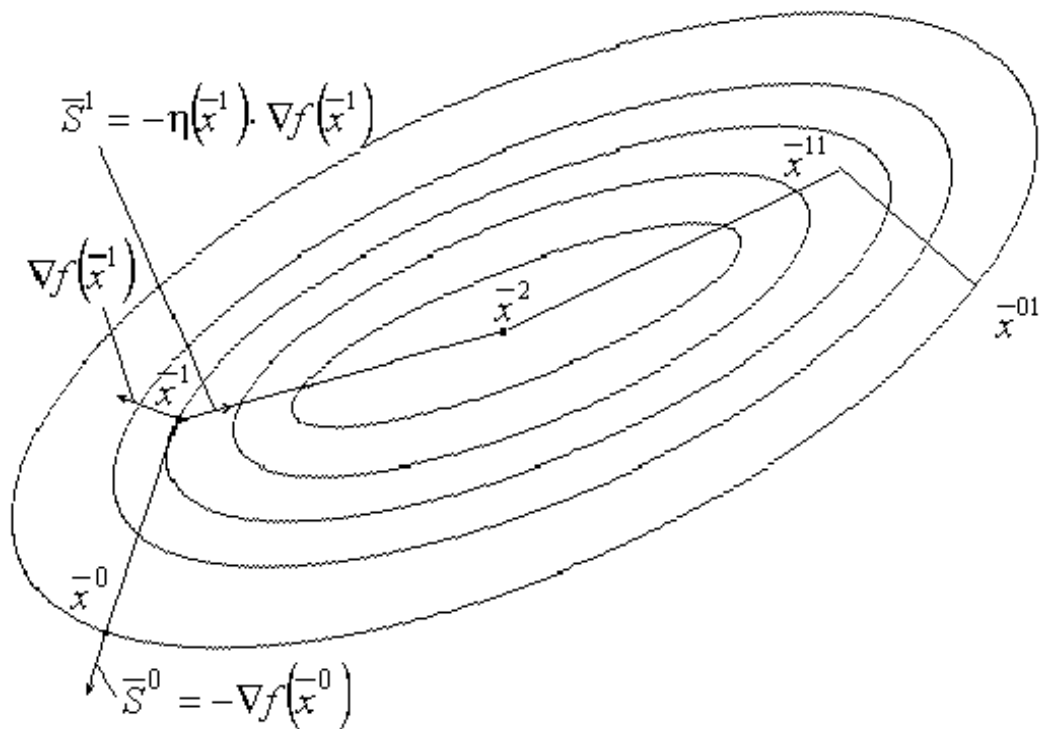


Рис.

2.Первая теорема двойственности.

(см. билет № 1, 2-ой вопрос)

13 билет

1.Алгоритм Дэвидона-Флетчер-Пауэлла.

Реальная разница алгоритмов метода переменной метрики заключается в нахождении $\Delta\eta$. В этом алгоритме матрица $\Delta\eta$ имеет ранг 2. Как и в предыдущем случае, матрица η перевычисляется т.о., чт. для квадратичной ф-ции после n шагов она совпала с матрицей

$H^{-1}(\bar{x}^k) = [\nabla^2 f(\bar{x}^k)]^{-1}$. Исходная матрица обычно выбирается единичной: $\eta^0 = E$. Хотя предпочтительней задание начального приближения элементов этой матрицы аналитическими значениями вторых частных производных или их конечно-разностными приближениями. Соотношение для $\Delta\eta^k$ в алгоритме Дэвидона-Флетчера-Пауэлла можно получить путем подстановки $\bar{y} = \Delta\bar{x}^k$, $\bar{z} = \eta^k \cdot \Delta\bar{g}^k$

в уравнение (6)
$$\left(\Delta\eta^k = \frac{1}{\omega} \cdot \frac{\Delta\bar{x}^k \cdot \bar{y}^T}{\bar{y}^T \cdot \Delta\bar{g}^k} - \frac{\eta^k \cdot \Delta\bar{g}^k \cdot \bar{z}^T}{\bar{z}^T \cdot \Delta\bar{g}^k} \right).$$

Тогда получим:

$$\eta^{k+1} = \eta^k + A^k - B^k = \eta^k + \frac{\Delta\bar{x}^k \cdot (\Delta\bar{x}^k)^T}{(\Delta\bar{x}^k)^T \cdot \Delta\bar{g}^k} - \frac{\eta^k \cdot \Delta\bar{g}^k \cdot (\Delta\bar{g}^k)^T \cdot (\eta^k)^T}{(\Delta\bar{g}^k)^T \cdot \eta^k \cdot \Delta\bar{g}^k}. \quad (8)$$

Следует отметить, что A^k и B^k являются симметрическими матрицами, так что η^k симметрическая и η^{k+1} будет симметрической.

Этот алгоритм явл-ся одним из наиболее эффективных алгоритмов переменной метрики. Алгоритм, использующий соотношение (8), оказывается достаточно эффективным при выполнении следующих условий:

- 1) Ошибки округления при вычислении $\nabla f(\bar{x}^k)$ невелики.
- 2) И матрица η^k в процессе вычислений не становится "плохой".

В ходе оптимизации этим методом происходит постепенный переход от градиентного направления спуска к ньютоновскому. При этом используются преимущества каждого метода на соответствующем этапе.

Роль матриц A^k в (8) заключается в обеспечении того, чт. $\eta \rightarrow H^{-1}$, тогда как матрица B^k обеспечивает положительную определенность матрицы η^{k+1} на всех этапах и в пределе их сумма исключает начальную матрицу η^0 . Используем (8) на нескольких этапах, начиная с η^0 :

$$\eta^1 = E + A^0 - B^0,$$

$$\eta^2 = \eta^1 + A^1 - B^1 = E + (A^0 + A^1) - (B^0 + B^1),$$

...

$$\eta^{k+1} = E + \sum_{i=0}^k A^i - \sum_{i=0}^k B^i$$

В случае квадратичной целевой ф-ции при $k = n - 1$ должно выполняться равенство $H^{-1} = \sum_{i=0}^{n-1} A^i$, а сумма матриц $\sum_{i=0}^k B^i$ строится т.о., чт. она сократилась с начальным значением η^0 .

При квадратичной ф-ции используемые направления поиска являются сопряженными. Именно это определяет эффективность алгоритма.

2.Вторая теорема двойственности.

(см. билет №2, вопрос 2-ой)

14 билет

1.Алгоритмы Пирсона. Проективный алгоритм Ньютона-Рафсона.

Ес. в выражении (6) $\left(\Delta \eta^k = \frac{1}{\omega} \cdot \frac{\Delta \bar{x}^k \cdot \bar{y}^T}{\bar{y}^T \cdot \Delta \bar{g}^k} - \frac{\eta^k \cdot \Delta \bar{g}^k \cdot \bar{z}^T}{\bar{z}^T \cdot \Delta \bar{g}^k} \right)$ положить

$\bar{y} = \bar{z} = \Delta \bar{x}^k$ и $\omega = 1$, тогда очередное приближение матрицы направлений определяется выражением: $\eta^{k+1} = \eta^k + \frac{[\Delta \bar{x}^k - \eta^k \cdot \Delta \bar{g}^k] \cdot (\Delta \bar{x}^k)^T}{(\Delta \bar{x}^k)^T \cdot \Delta \bar{g}^k}, \quad (9)$

где $\eta^0 = R^0$ – произвольная положительно определенная матрица.

Соответствующий алгоритм переменной метрики получил название **второго алгоритма Пирсона**. Метод обычно приводит к плохим направлениям поиска. Однако были примеры очень эффективного применения метода в приложениях.

Третий алгоритм Пирсона получается при подстановке в уравнение (6) следующих параметров: $\bar{y} = \bar{z} = \eta^k \cdot \Delta \bar{g}^k$ и $\omega = 1$. В этом случае итерационная ф-ла принимает вид:

$$\eta^{k+1} = \eta^k + \frac{[\Delta \bar{x}^k - \eta^k \cdot \Delta \bar{g}^k] \cdot [\eta^k \Delta \bar{g}^k]^T}{(\Delta \bar{g}^k)^T \eta^k \Delta \bar{g}^k}, \quad (10)$$

$$\eta^0 = R^0$$

Третий алгоритм на тестовых ф-циях оказывается более эффективным.

Проективный алгоритм Ньютона-Рафсона

Этот алгоритм также предложен Пирсоном. Получается из уравнения (6)

$$\text{при } \omega \rightarrow \infty \text{ и } \bar{z} = \eta^k \cdot \Delta \bar{g}^k : \quad \eta^{k+1} = \eta^k - \frac{[\eta^k \cdot \Delta \bar{g}^k] \cdot [\eta^k \cdot \Delta \bar{g}^k]^T}{(\Delta \bar{g}^k)^T \cdot \eta^k \cdot \Delta \bar{g}^k}. \quad (11)$$

$$\eta^0 = R^0$$

2.Метод последующего уточнения оценок.

(см. 3-ий билет, 2-ой вопрос)

15 билет

1.Метод Гринштадта и Гольдфарба.

Очередное приближение матрицы направлений определяется

$$\text{соотношением } \eta^{k+1} = \eta^k + \Delta \eta^k,$$

где в алгоритме Гринштадта –

$$\Delta \eta^k = \frac{1}{(\Delta \bar{g}^k)^T \cdot \eta^k \cdot \Delta \bar{g}^k} \cdot \left\{ \Delta \bar{x}^k \cdot (\Delta \bar{g}^k)^T \cdot \eta^k + \eta^k \cdot \Delta \bar{g}^k \cdot (\Delta \bar{x}^k)^T - \right. \\ \left. - \frac{\left[(\Delta \bar{g}^k)^T \Delta \bar{x}^k - (\Delta \bar{g}^k)^T \eta^k \Delta \bar{g}^k \right] \cdot \eta^k \Delta \bar{g}^k (\Delta \bar{g}^k)^T \eta^k}{(\Delta \bar{g}^k)^T \eta^k \Delta \bar{g}^k} \right\}, \quad (12)$$

а в алгоритме Гольдфарба –

$$\Delta \eta^k = \frac{1}{(\Delta \bar{g}^k)^T \cdot \eta^k \cdot \Delta \bar{g}^k} \cdot \left\{ \Delta \bar{x}^k \cdot (\Delta \bar{g}^k)^T \cdot \eta^k + \eta^k \cdot \Delta \bar{g}^k \cdot (\Delta \bar{x}^k)^T - \right. \\ \left. - \left[1 + \frac{(\Delta \bar{g}^k)^T \Delta \bar{x}^k}{(\Delta \bar{g}^k)^T \cdot \eta^k \cdot \Delta \bar{g}^k} \right] \cdot \eta^k \cdot \Delta \bar{g}^k \cdot (\Delta \bar{g}^k)^T \cdot \eta^k \right\}. \quad (13)$$

По эффективности данные методы сравнимы с алгоритмом Дэвидона-Флетчера-Пауэлла.

2.Метод потенциалов для решения транспортной задачи линейного программирования.

(см. 5-ый билет, 2-ой вопрос)

16 билет

1.Алгоритм Флетчера. Флетчером был предложен алгоритм, в котором условие окончания процесса для квадратичной ф-ции после n шагов

было отброшено, но сохранено свойство, заключающееся в том, что для квадратичных функций $\eta \rightarrow H^{-1}(\bar{x})$ в том смысле, что собственные значения η стремятся к собственным значениям H^{-1} .

Полученное Флетчером соотношение для очередного приближения матрицы η имеет вид:

$$\eta^{k+1} = \left[E - \frac{\Delta \bar{x}^k \cdot (\Delta \bar{g}^k)^T}{(\Delta \bar{x}^k)^T \cdot \Delta \bar{g}^k} \right] \cdot \eta^k \cdot \left[E - \frac{\Delta \bar{g}^k \cdot (\Delta \bar{x}^k)^T}{(\Delta \bar{x}^k)^T \cdot \Delta \bar{g}^k} \right] + \frac{\Delta \bar{x}^k \cdot (\Delta \bar{x}^k)^T}{(\Delta \bar{x}^k)^T \cdot \Delta \bar{g}^k}. \quad (14)$$

Однако в реализованном Флетчером алгоритме матрица η^k в зависимости от выполнения условий вычисляется по-разному:

$$\text{Ес. А) } (\Delta \bar{g}^k)^T \cdot H^{-1}(\bar{x}^k) \cdot \Delta \bar{g}^k < (\Delta \bar{g}^k)^T \cdot \eta^k \cdot \Delta \bar{g}^k;$$

то для вычисления очередного приближения η^{k+1} используется соотношение (8) (метод Дэвидона-Флетчера-Пауэлла), а ес.

$$\text{Б) } (\Delta \bar{g}^k)^T \cdot H^{-1}(\bar{x}^k) \cdot \Delta \bar{g}^k \geq (\Delta \bar{g}^k)^T \cdot \eta^k \cdot \Delta \bar{g}^k,$$

то используется соотношение (14).

Очевидно, что проверка условий (А-Б) затруднительна и теряет смысл, т.к. приходится находить $H^{-1}(\bar{x}^k)$. Поэтому при реализации алгоритма обычно используют ф-лу (14) без проверки условий (А-Б). Сравнение алгоритмов на тестовых ф-циях показывает, что в этом случае алгоритм Флетчера проигрывает по эффективности алгоритму ДФП.

Эффективность алгоритмов Дэвидона-Флетчера-Пауэлла и Флетчера в существенной мере определяют используемые методы одномерного поиска. Зачастую успех конкретной реализации определяет, именно, эффективность используемых методов одномерного поиска. В авторских реализациях методов ДФП и Флетчера решение задачи

$$\lambda^k = \arg \min f(\bar{x}^k + \lambda \bar{S}^k), \text{ где } \bar{S}^k = -\eta(\bar{x}^k) \cdot \nabla f(\bar{x}^k),$$

λ^k определяется на интервале $(0, \lambda')$ с помощью кубической

$$\text{интерполяции. Величина } \lambda' = \min \left\{ 1, \frac{2[f(\bar{x}^k) - f_0]}{\nabla^T f(\bar{x}^k) \bar{S}^k} \right\},$$

где f_0 – нижняя оценка значений $f(\bar{x})$, найденная в процессе

одномерного поиска по направлению \bar{S}^k (в процессе выделения интервала, содержащего минимум). Ес. найденное в результате кубической интерполяции значение λ оказывается больше λ' , то для пробных шагов начальное значение λ берется в виде $\lambda^{r+1} = 0,1\lambda^r$,

где r обозначает номер в по=>сти шагов при одномерном поиске. А ес.
 $\lambda < \lambda'$ и $f(\bar{x}^k + \lambda \bar{S}^k) < f_0$, то одномерный поиск заканчивается.

Вообще говоря, применение квадратичной интерполяции при одномерном поиске несколько не хуже кубической. И алгоритмы оказываются столь же эффективными.

2.Транспортная задача с ограничениями на пропускные способности: построение оптимального плана.

(см. 6-ой билет, 2-ой вопрос)

17 билет

1.Алгоритм Гольдштайна и Прайса.

Алгоритмы с аппроксимацией матрицы Гессе

Можно строить аналогичные алгоритмы, аппроксимируя в процессе поиска не матрицу $H^{-1}(\bar{x}^k)$, а матрицу $H(\bar{x}^k)$. А затем строить обратную к ней. Т.о., на каждой итерации находится $H(\bar{x}^{k+1}) \approx \Gamma^{k+1} = \Gamma^k + \Delta\Gamma^k$, где Γ^k – оценка $H(\bar{x}^k)$, а матрица $\Delta\Gamma^k$ – симметрическая матрица ранга 1, такая, что Γ^{k+1} удовлетворяет уравнению $\Gamma^{k+1} \Delta\bar{x}^k = \Delta\bar{g}^k$. При этом

$$\Delta\Gamma^k = \frac{[\Delta\bar{g}^k - \Gamma^k \cdot \Delta\bar{x}^k] \cdot [\Delta\bar{g}^k - \Gamma^k \cdot \Delta\bar{x}^k]^T}{[\Delta\bar{g}^k - \Gamma^k \cdot \Delta\bar{x}^k]^T \cdot \Delta\bar{x}^k}. \quad (15)$$

Поскольку в процессе поиска матрица $[\Gamma^{k+1}]^{-1}$ может не быть положительно определенной, следует в процессе поиска использовать ограничительные условия, обеспечивающие положительную определенность такой матрицы. В качестве начального приближения можно выбирать $\Gamma^0 = [\eta^0]^{-1}$.

К алгоритмам такого типа относится алгоритм **Гольштайна и Прайса**, который описывается следующей по=>стью действий и предназначен для минимизации выпуклых функций. Гольштайн и Прайс использовали не (15), а проводили аппроксимацию матрицы $H(\bar{x}^k)$ при помощи разностной схемы, основанной на полуфакториальном построении (полуфакториал – произведение n первых четных чисел), а затем проводили обращение матрицы. При этом для оценки $H(\bar{x}^k)$ требуется лишь информация о $f(\bar{x}^k)$ и $\nabla f(\bar{x}^k)$. На k -м этапе алгоритм выглядит следующим образом. Заранее заданы величины $0 < \delta < 1/2$ и $r > 0$.

1. Вычисляется в качестве аппроксимации $H(\bar{x}^k)$ матрица $\tilde{H}(\bar{x}^k)$, j -й столбец которой определяется по ф-ле $\nabla f(\bar{x}^k + \theta^k E_j) - \nabla f(\bar{x}^k)$,

где $\theta^k = r \|\varphi(\bar{x}^{k-1})\|$ для $k > 0$, $\theta^0 = r$, E_j – j -й столбец единичной матрицы E размерности $n \times n$. $\varphi(\bar{x}^k)$ – вектор-столбец, определяемый в соответствии с условиями:

- $\varphi(\bar{x}^k) = -\nabla f(\bar{x}^k)$, ес. $k = 0$ или $\tilde{H}(\bar{x}^k)$ сингулярна, или $\nabla^T f(\bar{x}^k) \tilde{H}^{-1}(\bar{x}^k) \nabla f(\bar{x}^k) \leq 0$, так что матрица $\tilde{H}^{-1}(\bar{x}^k)$ не явл-ся положительно определенной;
- $\varphi(\bar{x}^k) = -\tilde{H}^{-1}(\bar{x}^k) \nabla f(\bar{x}^k)$ – в противном случае.

Заметим, что матрица $\tilde{H}(\bar{x}^k)$ не обязательно симметрическая матрица, и ес. $\nabla^T f(\bar{x}^k) \tilde{H}^{-1}(\bar{x}^k) \nabla f(\bar{x}^k) \leq 0$, то предлагаемое направление поиска $\varphi(\bar{x}^k)$ и направление градиента $\nabla f(\bar{x}^k)$ отличаются более чем на 90° . \Rightarrow , $\varphi(\bar{x}^k)$ м.б. направлено в сторону, в которой $f(\bar{x})$ увеличивается.

2. Для выражения

$$F(\bar{x}^k, \lambda) = \frac{f(\bar{x}^k) - f(\bar{x}^k + \lambda \varphi(\bar{x}^k))}{\lambda [\nabla^T f(\bar{x}^k) \cdot \varphi(\bar{x}^k)]}$$

вычисляется λ^k так, чт. $\delta \leq F(\bar{x}^k, 1)$ или $\delta \leq F(\bar{x}^k, \lambda^k) \leq 1 - \delta$, $\lambda^k \neq 1$. Эти условия нужны для того, чт. не допускать шагов поиска, которые далеко выходят за область линейного изменения целевой ф-ции в окрестности \bar{x}^k , предполагавшуюся при аппроксимации $H(\bar{x})$.

3. Берется $\bar{x}^{k+1} = \bar{x}^k + \lambda^k \varphi(\bar{x}^k)$.

4. Процесс заканчивается, когда $\|\varphi(\bar{x}^k)\| < \varepsilon$.

Параметр r следует выбирать так, чт. матрица $\tilde{H}(\bar{x}^k)$ аппроксимировала $H(\bar{x}^k)$ как можно ближе. Величина δ выбирается так, чт. значения $f(\bar{x}^k)$, $k = 1, 2, \dots$, представляли собой монотонно убывающую по \Rightarrow сть; чем ближе значение δ к $1/2$, тем в большей степени $f(\bar{x}^k + \lambda \varphi(\bar{x}^k))$ приближается к своему минимуму по λ .

2. Транспортная задача с ограничениями на пропускные способности: построение опорного плана.

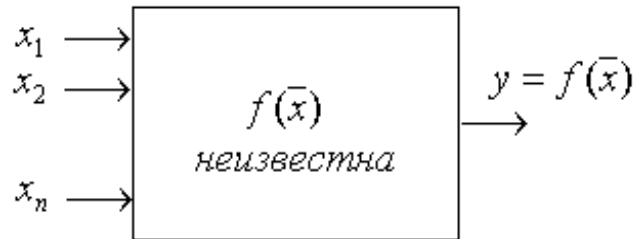
(см. 7-ой билет, 2-ой вопрос)

18 билет

1. Статистические методы локального поиска.

Статистические методы или *методы случайного поиска* получили достаточно широкое распространение при построении оптимальных решений в различных приложениях. Т. к.:

1) $C \uparrow$ размерности задач резко \downarrow эффективность регулярных методов поиска (детерминированных). 2) Часто информация об оптимизируемом объекте слишком мала для того, что можно было бы применить детерминированные методы. Достаточно часто статистические алг. исп-ют при поиске оптим. реш-я в сис-мах управления, когда отклик системы можно получить только при задании управляющих воздействий \bar{x} на ее входах. В таких ситуациях статистические алг-мы м. оказаться эффективнее детерминированных.



Наибольший эффект применение статистических методов приносит при решении задач большой размерности или при поиске глобального экстремума. *Под случайными или статистическими методами* поиска будем понимать методы, использующие элемент случайности либо при сборе информации о целевой ф-ции при пробных шагах, либо для улучшения значений ф-ции при рабочем шаге. Случайным образом может выбираться направление спуска, длина шага, величина штрафа при нарушении ограничения и т.д.

Статистические алгоритмы обладают рядом достоинств:

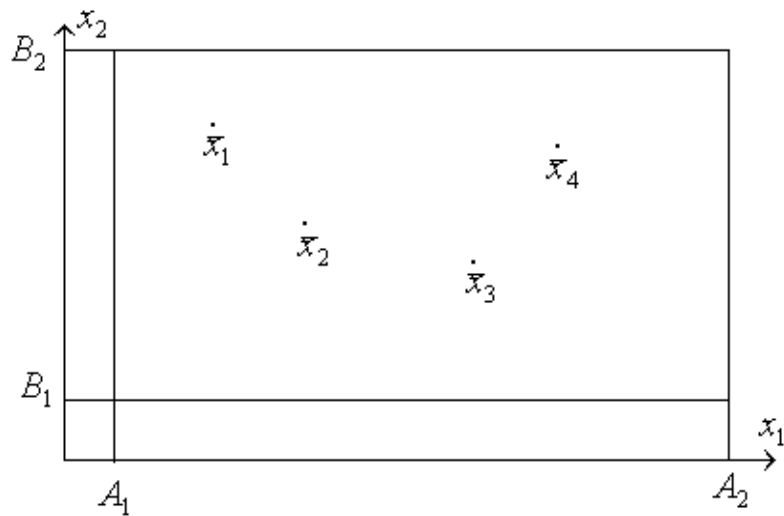
- простота реализации и отладки программ;
- надежность и помехоустойчивость;
- универсальность;
- возможность введения операций обучения в алгоритм поиска;
- возможность введения операций прогнозирования оптимальной точки (оптимального решения).

Недостатками явл. большое кол-во вычислений минимизируемой ф-ции и медленная сх-сть в районе экстремума. Принято считать, что преимущество статистических методов проявл-ся с ростом размерности задач

Простой случайный поиск

Пусть нам необх. решить задачу минимизации ф-ции $f(\bar{x})$ при условии, что $\bar{x} \in [\bar{A}, \bar{B}]$. В данной области по равномерному закону выбираем случайную точку \bar{x}_1 и вычисляем в ней значение ф-ции $y_1 = f(\bar{x}_1)$. Затем выбираем таким же образом случайную точку \bar{x}_2 и вычисляем $y_2 = f(\bar{x}_2)$. Запоминаем минимальное из этих значений и точку, в которой значение ф-ции минимально. Далее генерируем новую точку. Делаем N экспериментов, после чего лучшую точку берем в

качестве решения задачи (в которой ф-ция имеет минимальное значение среди всех случайно сгенерированных).



Оценим число экспериментов, необх.е для определения решения (точки минимума) с заданной точностью. Пусть n - размерность вектора переменных. Объем n -мерного прямоугольника, в котором ведется поиск минимума, $V = \prod_{i=1}^n (B_i - A_i)$. Ес. необх. найти решение с точностью ε_i ,

$i = \overline{1, n}$, по каждой из переменных, то мы должны попасть в окрестность оптимальной точки с объемом $V_\varepsilon = \prod_{i=1}^n \varepsilon_i$. Вероятность попадания в эту

окрестность при одном испытании равна $P_\varepsilon = \frac{V_\varepsilon}{V}$. Вероятность

непопадания равна $1 - P_\varepsilon$. Испытания независимы, поэтому вероятность непопадания за N экспериментов равна $(1 - P_\varepsilon)^N$.

Вер-ть, что мы найдем решение за N испытаний: $P = 1 - (1 - P_\varepsilon)^N$. Отсюда нетрудно получить оценку необх.го числа испытаний N для определения минимума с требуемой точностью: $N \geq \frac{\ln(1 - P)}{\ln(1 - P_\varepsilon)}$.

Различают направленный и ненаправленный случайный поиск.

1. **Ненаправленный случайный поиск.** При таком поиске все последующие испытания проводят совершенно независимо от результатов предыдущих. Сходимость такого поиска очень мала, но имеется важное преимущество, связанное с возможностью решения многоэкстремальных задач (искать глобальный экстремум). Примером явл-ся рассмотренный простой случайный поиск.
2. **Направленный случайный поиск.** В этом случае отдельные испытания связаны м/у собой. Результаты проведенных испытаний используются для

формирования последующих. Сходимость таких методов, как правило, выше, но сами методы обычно приводят только к локальным экстремумам.

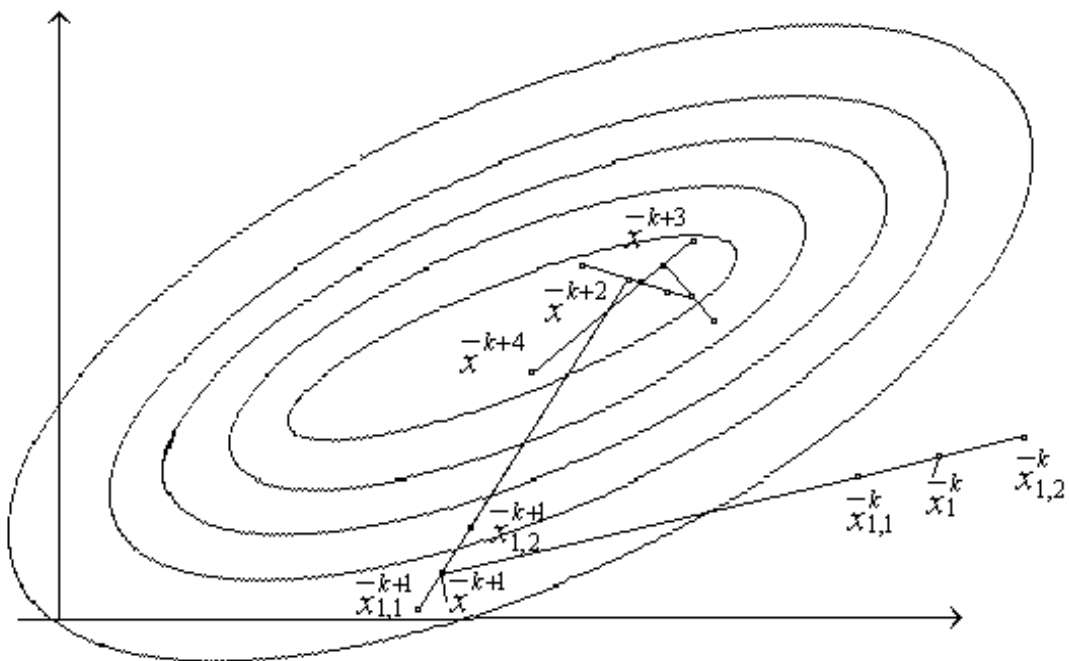
Простейшие алгоритмы направленного случайного поиска

Алгоритм парной пробы. В данном алгоритме четко разделены пробный и рабочий шаги.

Пусть \bar{x}^k – найденное на k -м шаге наименьшее значение минимизируемой ф-ции $f(\bar{x})$. По равномерному закону генерируется случайный единичный вектор $\bar{\xi}$ и по обе стороны от исходной точки \bar{x}^k делаются две пробы: проводим вычисление ф-ции в точках $\bar{x}_{1,2}^k = \bar{x}^k \pm g \cdot \bar{\xi}$, где g - величина пробного шага.

Рабочий шаг делается в направлении наименьшего значения целевой ф-ции. Очередное приближение определяется соотношением

$$\bar{x}^{k+1} = \bar{x}^k + \Delta \bar{x}^k = \bar{x}^k + a \cdot \bar{\xi} \cdot \text{sign} \left(f(\bar{x}^k - g \bar{\xi}) - f(\bar{x}^k + g \bar{\xi}) \right).$$



Особенностью данного алгоритма явл-ся его повышенная тенденция к “блужданию”. Даже найдя экстремум, алгоритм может увести процесс поиска в сторону.

Алгоритм наилучшей пробы. На k -м шаге мы имеем точку \bar{x}^k .

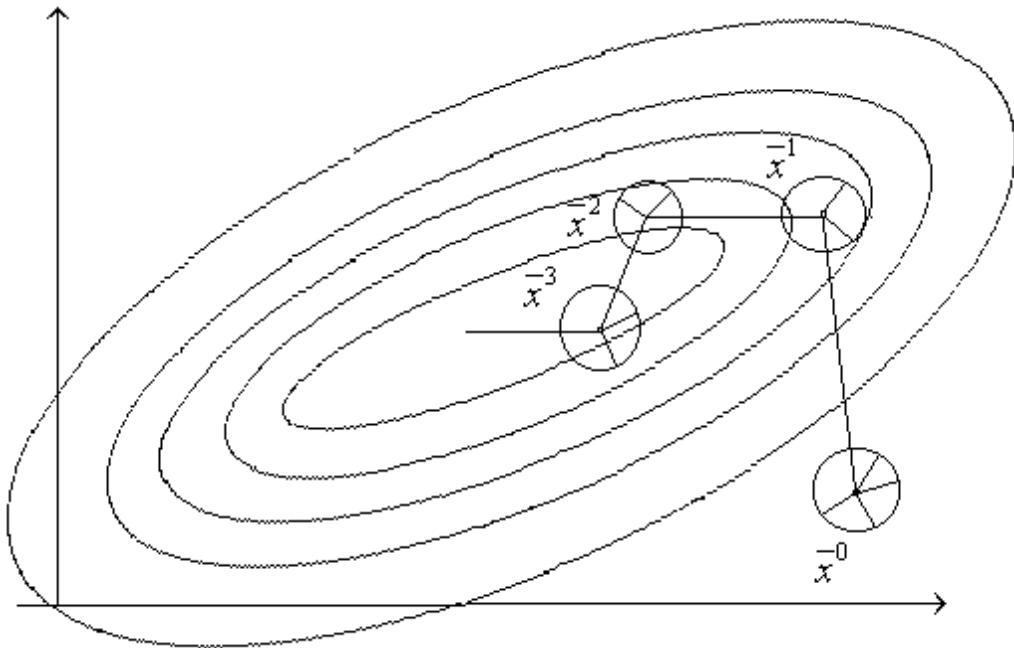
Генерируется m случайных единичных векторов $\bar{\xi}_1, \dots, \bar{\xi}_m$. Делаются пробные шаги в направлениях $g \cdot \bar{\xi}_1, \dots, g \cdot \bar{\xi}_m$ и в точках

$\bar{x}^k + g \cdot \bar{\xi}_1, \dots, \bar{x}^k + g \cdot \bar{\xi}_m$ вычисляются значения ф-ции. Выбирается тот

шаг, который приводит к наибольшему уменьшению ф-ции:

$\bar{\xi}^* = \arg \min_{i=1,m} f(\bar{x}^k + g \cdot \bar{\xi}_i)$. В данном направлении дел-ся шаг $\Delta \bar{x}^k = \lambda \cdot \bar{\xi}^*$.

Параметр λ м.определяться как результат минимизации по направлению, определяемому наилучшей пробой, или выбираться по определенному закону. С увеличением числа проб выбранное направление приближается к направлению $-\nabla f(\bar{x})$. Ес. ф-ция $f(\bar{x})$ близка к линейной, то есть возможность ускорить поиск, выбирая вместе с наилучшей и наихудшую пробу. Тогда рабочий шаг можно делать или в направлении наилучшей, или в направлении, противоположном наихудшей пробе.



Метод статистического градиента. Из исходного состояния \bar{x}^k делается m независимых проб $g \cdot \bar{\xi}_1, \dots, g \cdot \bar{\xi}_m$ в m случайных направлениях, а затем вычисляются соответствующие значения минимизируемой ф-ции в этих точках. Для каждой пробы запоминаем приращения ф-ции

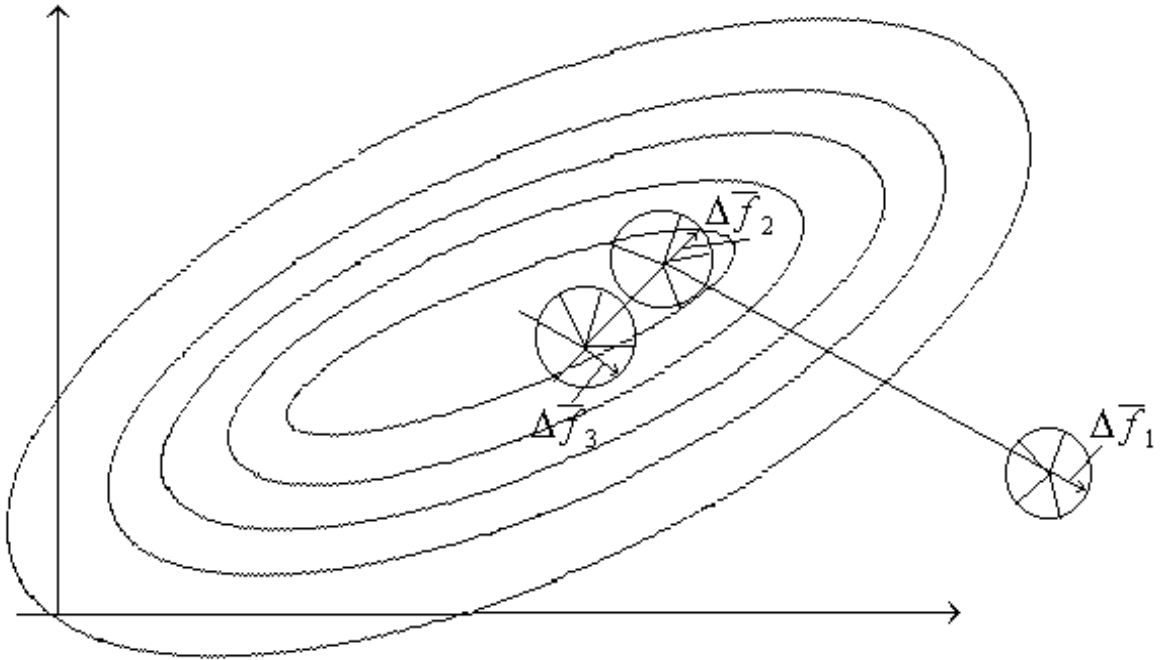
$$\Delta f_j = f(\bar{x}^k + g \cdot \bar{\xi}_j) - f(\bar{x}^k).$$

После этого формируем векторную сумму $\Delta \bar{f} = \sum_{j=1}^m \bar{\xi}_j \cdot \Delta f_j$.

В пределе при $m \rightarrow \infty$ направление $\Delta \bar{f}$ совпадает с направлением градиента целевой ф-ции. При конечном m вектор $\Delta \bar{f}$ представ.собой статистическую оценку направления градиента. В направлении $\Delta \bar{f}$ делается рабочий шаг и, в результате, очередное приближение

определяется соотношением $\bar{x}^{k+1} = \bar{x}^k - \lambda \cdot \frac{\Delta \bar{f}}{\|\Delta \bar{f}\|}$.

При выборе оптимального значения λ , которое минимизирует функцию в заданном направлении, мы получаем статистический вариант метода наискорейшего спуска. Существенное преимущество перед детерминированными алгоритмами заключается в возможности принятия решения о направлении рабочего шага при $m < n$. При $m = n$ и неслучайных ортогональных рабочих шагах, направленных вдоль осей координат, алгоритм вырождается в градиентный метод.



2. Двойственность задачи линейного программирования.

(см. 24-ый билет, 1-ый вопрос)

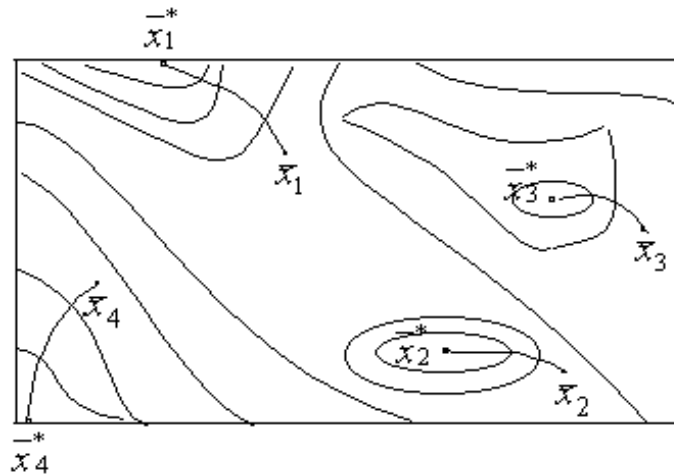
19 билет

1. Алгоритмы глобального поиска.

Случ. поиск приобретает решающее знач-е при реш-и многоэкстремальных задач и оптимизации сложных объектов. В общ. случае реш-е многоэкстр-х задач без элемента случайности практически невозможно. Рассмотрим некоторые подходы к поиску глобального экстремума.

Алгоритм 1. В допустимой области D случайным образом выбирают точку $\bar{x}_1 \in D$. Приняв эту точку за исходную и используя некоторый детерминированный метод или алгоритм направленного случайного поиска, осуществляется спуск в точку локального минимума $\bar{x}_1^* \in D$, в области притяжения которой оказалась точка \bar{x}_1 .

Затем выбирается новая случайная точка $\bar{x}_2 \in D$ и по той же схеме осуществляется спуск в точку локального минимума $\bar{x}_2^* \in D$, и т.д.



Поиск прекращается, как только некоторое заданное число m раз не удастся найти точку локального экстремума со значением Φ -ции, меньшим предыдущих.

Алгоритм 2. Пусть получена некоторая точка локального экстремума $\bar{x}_1^* \in D$. После этого переходим к *ненаправленному случайному* поиску до получения точки \bar{x}_2 такой, что $f(\bar{x}_2) < f(\bar{x}_1^*)$.

Из точки \bar{x}_2 с помощью детерминированного алгоритма или направленного случайного поиска получаем точку локального экстремума \bar{x}_2^* , в которой заведомо выполняется неравенство $f(\bar{x}_2^*) < f(\bar{x}_1^*)$.

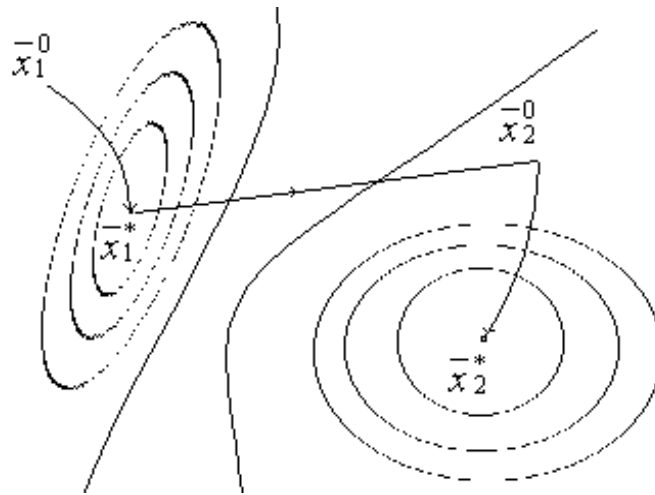
Далее с помощью случайного поиска определяем новую точку \bar{x}_3 , для которой справедливо неравенство $f(\bar{x}_3) < f(\bar{x}_2^*)$, и снова спуск в точку локального экстремума \bar{x}_3^* , и т.д.

Поиск прекращается, если при генерации некоторого предельного числа новых случайных точек не удастся найти лучшей, чем предыдущий локальный экстремум, который тогда и принимается в качестве решения.

Алгоритм 3. Пусть \bar{x}_1^0 — некоторая исходная точка поиска в области D , из которой осуществляется спуск в точку локального экстремума \bar{x}_1^* со значением $f(\bar{x}_1^*)$. Далее из точки \bar{x}_1^* движемся либо в случайном направлении, либо в направлении $\bar{x}_1^0 - \bar{x}_1^*$ до тех пор, пока Φ -ция снова не начнет убывать (выходим из области притяжения \bar{x}_1^*).

Полученная точка \bar{x}_2^0 принимается за начало следующего спуска. В результате находим новый локальный экстремум \bar{x}_2^* со значением Φ -ции $f(\bar{x}_2^*)$.

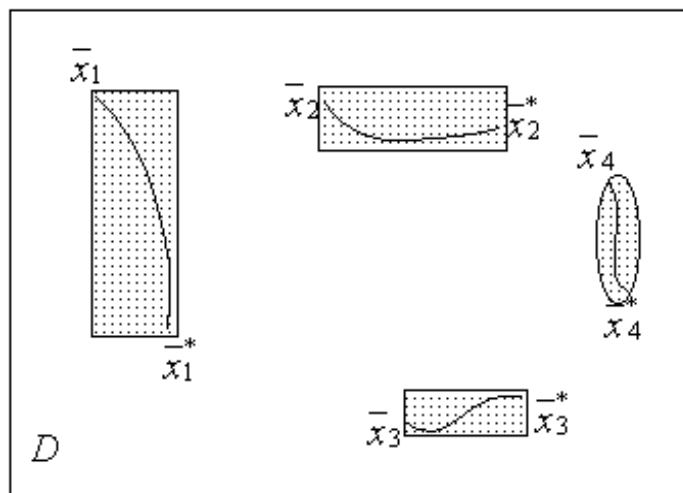
Если $f(\bar{x}_2^*) < f(\bar{x}_1^*)$, точка \bar{x}_1^* забывается и ее место занимает точка \bar{x}_2^* .
Если $f(\bar{x}_2^*) \geq f(\bar{x}_1^*)$, то возвращаемся в точку \bar{x}_1^* и движемся из нее в новом случайном направлении.



Процесс прекращ,ес. не удастся найти лучший лок.минимум после заданного числа попыток или не удастся найти “случайного” направления, в котором ф-ция снова начинает убывать.Такой подход позволяет найти глобальный экстремум в случае многосвязных допустимых областей.

Алгоритм 4. В допустимой области D разбрасываем m случайных точек и выбираем из них наилучшую, то есть ту, в которой значение ф-ции минимально. Из выбранной точки осуществляем локальный спуск. А далее вокруг траектории спуска образуем *запретную область*.

В оставшейся области случайным образом разбрасываем новую совокупность случайных точек и из лучшей точки осуществляем спуск в точку локального экстремума. Вокруг новой траектории также строим запретную область и т.д.



Поиск прекращается, ес. в течение заданного числа попыток не удастся найти лучшего локального экстремума.**Замечание:** Комбинация случайного поиска с детерминированными методами применяется не только для решения многоэкстремальных задач. Часто к такой комбинации прибегают в ситуациях, когда детерминированные методы сталкиваются с теми или иными трудностями (застревают на дне узкого оврага, в седловой точке и т.п.). Шаг в случайном направлении порой позволяет преодолеть такую тупиковую ситуацию для детерминированного алгоритма.

2. Задача о максимальном потоке в транспортной сети.

(см. 9-ый билет, 2-ой вопрос)

20 билет

1. Методы штрафных функций.

Наиболее распространенные задачи на практике это оптимизационные задачи при наличии ограничений, т.е. задачи поиска оптимального решения, удовлетворяющего некоторой системе ограничений. Существование эффективных алгоритмов решения безусловных задач оптимизации всегда толкает на попытку использования этих методов для решения условных задач после соответствующего преобразования условной задачи к некоторой эквивалентной ей безусловной задаче. Пусть необх. решить задачу

$$\min \left\{ f(\bar{x}) \mid h_j(\bar{x}) = 0, \quad j = \overline{1, m}; \quad g_j(\bar{x}) \leq 0, \quad j = \overline{m+1, k} \right\}, \quad (1)$$

в которой целевая ф-ция и ф-ции системы ограничений представляют собой выпуклые ф-ции (желательно). Основная идея метода штрафных функций заключается в следующем. Строят такую вспомогательную

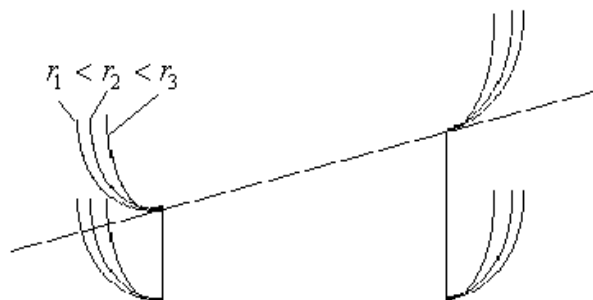
$$\text{функцию } Q(\bar{x}, \bar{r}) = f(\bar{x}) + \sum_{j=1}^m r_j \cdot H[h_j(\bar{x})] + \sum_{j=m+1}^k r_j \cdot G[g_j(\bar{x})], \quad (2)$$

что приближенное решение задачи (1) получалось в результате решения последовательности задач безусловной минимизации функций (2)

$$\min Q(\bar{x}, \bar{r}). \quad (3)$$

В методе (внешних) **штрафных функций** ф-ции $H(\cdot)$ и $G(\cdot)$ выбираются т.о., что они становились отличными от нуля (положительными) при нарушении соответствующего ограничения. А т.к. мы минимизируем (2), то движение в сторону нарушения ограничения становится невыгодным. В данном методе ф-ции $H(\cdot)$ и $G(\cdot)$ внутри допустимой области должны быть равны нулю. Например, для ограничений неравенств: $G_j[g_j(\bar{x})] \rightarrow 0$ при $g_j(\bar{x}) \rightarrow 0^+$.

Приближенное решение задачи (1) получается в результате решения последовательности задач (3) при $r_j \rightarrow \infty, \quad j = \overline{1, k}$. Соответствующие методы называют еще *методами внешней точки*.



В методе *барьерных функций* ф-ции $H(\cdot)$ и $G(\cdot)$ в допустимой области выбираются отличными от нуля и такими, чт. при приближении к границе допустимой области (изнутри) они возрастают, препятствуя выходу при поиске за границу области. В этом случае эти ф-ции должны быть малыми (*положительными или отрицательными*) внутри допустимой области и большими *положительными* вблизи границы (внутри области). Например, для ограничений неравенств:

$$G_j[g_j(\bar{x})] \rightarrow \infty \text{ при } g_j(\bar{x}) \rightarrow 0^-.$$

Такие методы называют ещё *методами внутренней точки*. В алгоритмах, использующих ф-ции штрафа данного типа (барьерные ф-ции), требуют, чт. в процессе поиска точка \bar{x} всегда оставалась внутренней точкой допустимой области. Приближенное решение задачи (1) получается в результате решения по=>сти задач вида (3) при $r_j \rightarrow 0$, $j = \overline{1, k}$.

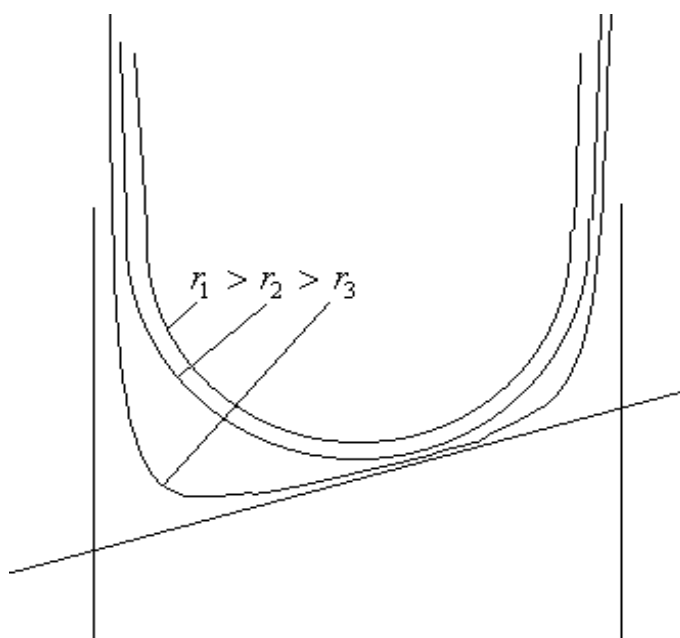


Рис.

При выборе функций штрафов для ограничений равенств обычно требуют, чт. $H_j[h_j(\bar{x})] \rightarrow 0$ при $h_j(\bar{x}) \rightarrow 0$.

Это могут быть, например ф-ции следующего вида:

- 1) $H_j[h_j(\bar{x})] = |h_j(\bar{x})|$,
- 2) $H_j[h_j(\bar{x})] = |h_j(\bar{x})|^2$,
- 3) $H_j[h_j(\bar{x})] = |h_j(\bar{x})|^\alpha$, при четном α .

Для ограничений неравенств ф-ции штрафа подбирают т.о., чт.

$$G_j[g_j(\bar{x})] = 0, \text{ при } g_j(\bar{x}) \leq 0; G_j[g_j(\bar{x})] > 0, \text{ при } g_j(\bar{x}) > 0.$$

Этому требованию отвечают, например, ф-ции вида:

- 1) $G_j[g_j(\bar{x})] = \frac{1}{2} \{ g_j(\bar{x}) + |g_j(\bar{x})| \},$
- 2) $G_j[g_j(\bar{x})] = \left[\frac{1}{2} \{ g_j(\bar{x}) + |g_j(\bar{x})| \} \right]^2,$
- 3) $G_j[g_j(\bar{x})] = \left[\frac{1}{2} \{ g_j(\bar{x}) + |g_j(\bar{x})| \} \right]^\alpha,$ при четном α .

В качестве барьерных функций для ограничений неравенств могут служить, например, ф-ции вида:

- 1) $G_j[g_j(\bar{x})] = -\frac{1}{g_j(\bar{x})},$ 2) $G_j[g_j(\bar{x})] = -\ln[-g_j(\bar{x})].$

По-прежнему действий при реализации методов штрафных или барьерных функций выглядит следующим образом:

1. На основании задачи (1) строим функцию (2). Выбираем начальное приближение \bar{x} и начальные значения коэффициентов штрафа r_j .
2. Решаем задачу (3).
3. Ес. полученное решение не удовлетворяет системе ограничений, то в случае использования метода штрафных функций увеличиваем значения коэффициентов штрафа r_j и снова решаем задачу (3). В случае метода барьерных функций значения коэффициентов r_j уменьшаются, чт. можно было получить решение на границе.
4. Процесс прекращается, ес. найденное решение удовлетворяет системе ограничений с определенной точностью.

2.Метод потенциалов для решения транспортной задачи с ограничениями на пропускные способности.

(см. 6-ой билет, 2-ой вопрос)

21 билет

1.Основные определения и теоремы линейного прогр-я.

Определение 1. Задача, в которой требуется минимизировать (или

максимизировать) линейную форму $\sum_{i=1}^n c_i \cdot x_i \rightarrow \min (\max)$

при усл, что $\sum_{i=1}^n a_{ij} \cdot x_i \leq b_j, j = \overline{1, m}$, или $\sum_{i=1}^n a_{ij} \cdot x_i = b_j, j = \overline{(m+1), p}$, и

$$x_i > 0, i = \overline{1, n},$$

наз-ся задачей лин-го программирования в произвольной форме записи.

Тогда задачу лин.прогр-я можно записать в виде: $\sum_{i=1}^n c_i \cdot x_i \rightarrow \min (\max)$

$$x_1 \cdot \bar{A}_1 + x_2 \cdot \bar{A}_2 + \dots + x_n \cdot \bar{A}_n + x_{n+1} \cdot \bar{A}_{n+1} + \dots + x_{n+m} \cdot \bar{A}_{n+m} = A_0, \bar{x} \geq \bar{0}.$$

Векторы \bar{A}_i называются векторами условий, а сама задача линейного программирования называется расширенной по отношению к исходной.

Пусть D и D_1 - допустимые мн-ва решений исходной и расширенной задач линейного программирования соответственно. Тогда любой точке множества D_1 соответствует единственная точка множества D и наоборот. В общем случае, допустимое мн-во D исходной задачи есть проекция множества D_1 расширенной задачи на подпр-во исходных переменных.

Определение 4. Набор чисел $\bar{x} = [x_1, x_2, \dots, x_n]^T$, удовлетворяющий ограничениям задачи линейного программирования, называется ее **планом**.

Определение 5. Реш-ем задачи линейного программирования называется ее план, минимизирующий (или максимизирующий) линейную форму.

Введем понятие базисного решения. Из матрицы расширенной задачи $A_p = [\bar{A}_1, \bar{A}_2, \dots, \bar{A}_{n+m}]$ выберем m линейно независимых векторов-столбцов, которые обозначим как матрицу $B_{m \times m}$, а ч/з $D_{m \times n}$ – обозначим матрицу из оставшихся столбцов. Тогда $A_p = [B, D]$, и ограничения расширенной задачи линейного программирования можно записать в виде:

$$A_p \cdot \bar{x} = B \cdot \bar{x}_B + D \cdot \bar{x}_D = \bar{A}_0. \quad (3)$$

Очевидно, что столбцы матрицы B образуют базис m -мерного пространства. Поэтому вектор \bar{A}_0 и любой столбец матрицы D можно представить в виде линейной комбинации столбцов матрицы B .

$$\text{Умножим (3) на } B^{-1} \text{ слева } B^{-1} \cdot B \cdot \bar{x}_B + B^{-1} \cdot D \cdot \bar{x}_D = B^{-1} \cdot \bar{A}_0, \quad (4)$$

$$\text{и найдем отсюда } \bar{x}_B: \bar{x}_B = B^{-1} \cdot \bar{A}_0 - B^{-1} \cdot D \cdot \bar{x}_D. \quad (5)$$

Придавая \bar{x}_D различные значения, будем получать различные решения \bar{x}_B .

$$\text{Ес. положить } \bar{x}_D = \bar{0}, \text{ то } \bar{x}_B = B^{-1} \cdot \bar{A}_0. \quad (6)$$

Реш.(6) наз-ют **базисным реш-ем** сис-мы из m ур-ний с $m+n$ неизв-ми.

Ес. полученное решение содержит только положительные компоненты, то оно называется **базисным допустимым**.

Особенность допустимых базисных решений состоит в том, что они явл. крайними точками допустимого множества D_1 расширенной задачи. Ес. среди компонент \bar{x}_B нет нулевых, то базисное допустимое реш-е наз. **невыврожденным**.

Определение 6. План \bar{x} задачи линейного программирования будем называть опорным, ес. векторы условий \bar{A}_i с положительными коэффициентами линейно независимы.

То есть, опорный план – это базисное допустимое решение расширенной системы, угловая точка многогранника решений.

Определение 7. Опорное решение называется **невыврожденным**, ес. оно содержит m положительных компонент (по числу ограничений).

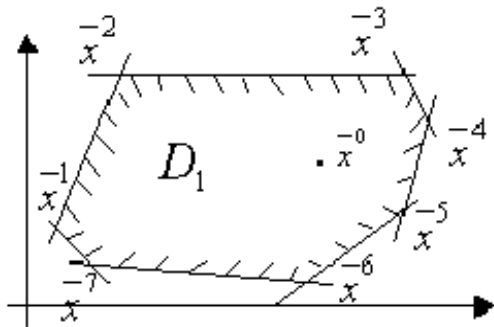
Невырожденный опорный план образуется пересечением n гиперплоскостей из образующих допустимую область. В случае вырожденности в угловой точке многогранника решений пересекается более чем n гиперплоскостей.

Теорема 1 (основная теорема линейного программирования).

- 1) Линейная форма $z = \bar{c}^T \cdot \bar{x}$ достигает своего минимума в угловой точке многогранника решений.
- 2) Ес. она принимает минимальное решение более чем в одной угловой точке, то она достигает того же самого значения в любой точке, являющейся выпуклой комбинацией этих угловых точек.

Док-во: Док-во теоремы основано на следующей лемме.

Лемма. Ес. D - замкнутое, ограниченное, выпуклое мн-во, имеющее конечное число крайних (угловых) точек, то любая точка $\bar{x} \in D$ м.б. представлена в виде выпуклой комбинации крайних точек D .



1) Пусть \bar{x}^0 - некоторая внутренняя точка. Многогранник ограниченный замкнутый, имеет конечное число угловых точек. D – допустимое мн-во.

Предположим, что точка \bar{x}^0 явл-ся оптимальной точкой. То есть, $z(\bar{x}^0) \leq z(\bar{x})$,

$\forall \bar{x} \in D$. Предположим, что точка \bar{x}^0 не явл-ся угловой. Тогда на основании леммы точку \bar{x}^0 можно выразить ч/з угловые точки многогранника \bar{x}^i , т.е. $\bar{x}^0 = \sum_{i=1}^p \alpha_i \cdot \bar{x}^i$, $\forall \alpha_i \geq 0$, $\sum_{i=1}^p \alpha_i = 1$.

Т.к. ф-ция $z(\bar{x})$ линейна, то $z(\bar{x}^0) = \sum_{i=1}^p \alpha_i \cdot z(\bar{x}^i)$. (*)

Выберем среди точек \bar{x}^i ту, в которой линейная форма $z(\bar{x})$ принимает наименьшее значение. Пусть это будет точка \bar{x}^k . Обозначим минимальное значение ф-ции в угловой точке ч/з z^* :

$$z^* = z(\bar{x}^k) = \min_{1 \leq i \leq p} \{z(\bar{x}^1), z(\bar{x}^2), \dots, z(\bar{x}^p)\}.$$

Подставим данное значение ф-ции в линейную форму (*) вместо $z(\bar{x}^i)$ и получим: $z(\bar{x}^0) \geq \sum_{i=1}^p \alpha_i \cdot z^* = z^* \cdot \sum_{i=1}^p \alpha_i = z^*.$

Т.к. \bar{x}^0 - оптимальная точка, то получили противоречие: $z(\bar{x}^0) \geq z^* (!).$ \Rightarrow , $z(\bar{x}^0) = z(\bar{x}^k)$, $\bar{x}^0 = \bar{x}^k$ - угловая точка.

2) Предположим, что линейная форма $z(\bar{x})$ принимает минимальное значение более чем в одной угловой точке, например, в угловых точках $\bar{x}^1, \bar{x}^2, \dots, \bar{x}^q$ $z(\bar{x}^1) = z(\bar{x}^2) = \dots = z(\bar{x}^q) = z^*$. Тогда ес. \bar{x} явл-ся выпуклой комбинацией этих точек, то есть $\bar{x} = \sum_{i=1}^q \alpha_i \cdot \bar{x}^i$, $\sum_{i=1}^q \alpha_i = 1$ и $\forall i \alpha_i \geq 0$,

$$\text{то } z(\bar{x}) = z\left(\sum_{i=1}^q \alpha_i \cdot \bar{x}^i\right) = z^* \cdot \sum_{i=1}^q \alpha_i = z^*.$$

То есть, ес. минимальное значение достигается более чем в одной угловой точке, то того же самого значения линейная форма достигает в любой точке, являющейся выпуклой комбинацией этих угловых точек

Теорема 2. Ес. известно, что системы векторов условий $\bar{A}_1, \dots, \bar{A}_m$, ($m \leq n$) линейно независима и такова, что $x_1 \cdot \bar{A}_1 + \dots + x_m \cdot \bar{A}_m = \bar{A}_0$,

где все $x_j > 0$, то точка $\bar{x} = [x_1, \dots, x_m, 0, \dots, 0]^T$ явл-ся угловой точкой многогранника решений.

Теорема 3. Ес. вектор \bar{x} явл-ся угловой точкой многогранника решений, то векторы условий, соответствующие положительным компонентам вектора \bar{x} , являются линейно независимыми.

Следствия: 1) Угловая точка многогранника решений имеет не более m положительных компонент вектора \bar{x} .

2) Каждой угловой точке многогранника решений соответствует m линейно независимых векторов из данной системы: $\bar{A}_1, \dots, \bar{A}_n$.

2. Методы штрафных функций. (20-ый билет, 1-ый вопрос)

22 билет

1. Симплекс-метод.

Его называют еще **методом по=>го улучшения плана**. Метод предназначен для решения общей задачи линейного программирования.

Введение в симплекс-метод

Пусть имеем след.задачу: $Q(\bar{x}) = c_1 \cdot x_1 + c_2 \cdot x_2 + \dots + c_n \cdot x_n \rightarrow \min$, (1)

с системой ограничений следующего вида:

[illegible]

Разрешим эту систему относительно переменных x_1, \dots, x_m :

[illegible]

Векторы условий, соответствующие x_1, \dots, x_m , образуют базис. Переменные x_1, \dots, x_m назовем базисными переменными. Остальные переменные задачи – небазисные.

Целевую функцию можно выразить ч/з небазисные переменные:

$$Q(\bar{x}) = c'_{m+1} \cdot x_{m+1} + c'_{m+2} \cdot x_{m+2} \dots + c'_n \cdot x_n + c'_0 \rightarrow \min.$$

Ес. приравнять небазисные переменные нулю

$$x_{m+1} = 0, \quad x_{m+2} = 0, \quad \dots, \quad x_n = 0,$$

то соответствующие базисные переменные примут значения

$$x_1 = b_1', \quad x_2 = b_2', \quad \dots, \quad x_m = b_m'.$$

В-р \bar{x} с такими компонентами предст-т собой угловую точку многогранника реш-й(допустимую)при усл., что $b'_i \geq 0$ (опорный план).

Теперь необх. перейти к другой угловой точке с меньшим значением целевой ф-ции. Для этого следует выбрать некоторые небазисную и базисную переменные так, чт. после того, как мы “поменяем их местами”, значение целевой ф-ции уменьшится. Такой направленный перебор в конце концов приведет нас к решению задачи.

Алгоритм симплекс метода

Формализованный алгоритм симплекс метода состоит из двух основных этапов: 1) построение опорного плана; 2) построение оптимального плана.

Проиллюстрируем алгоритм на рассмотренном ранее примере:

$$Q(\bar{x}) = x_4 - x_5 \rightarrow \min$$

$$\left. \begin{aligned} x_1 + x_4 - 2x_5 &= 1 \\ x_2 - 2x_4 + x_5 &= 2 \\ x_3 + 3x_4 + x_5 &= 3 \end{aligned} \right\}.$$

$$\bar{x} \geq \bar{0}.$$

В случае базисных переменных $\{x_1, x_2, x_3\}$ начальная симплексная таблица для данного примера будет выглядеть следующим образом:

	$-x_4$	$-x_5$	1
$x_1 =$	1	-2	1
$x_2 =$	-2	1	2
$x_3 =$	3	1	3
$Q(\bar{x})$ =	-1	1	0

Она уже соответствует опорному плану $\bar{x}^1 = [1 \ 2 \ 3 \ 0 \ 0]^T$ (столбец свободных членов).

Построение оптимального плана. Для того чт. опорный план был оптимальен при минимизации целевой ф-ции необх., чт. коэффициенты в строке целевой ф-ции были неположительными (в случае максимизации – неотрицательными). То есть, при поиске минимума мы должны освободиться от положительных коэффициентов в строке $Q(\bar{x})$.

Выбор разрешающего элемента. Ес. при поиске минимума в строке целевой ф-ции есть коэффициенты больше нуля, то выбираем столбец с положительным коэффициентом в строке целевой ф-ции в качестве разрешающего. Пусть это столбец с номером l .

Для выбора разрешающей строки (разрешающего элемента) среди положительных коэффициентов разрешающего столбца выбираем тот (ту строку), для которого отношение коэффициента в столбце свободных членов к коэффициенту в разрешающем столбце минимально:

$$\frac{b_r}{a_{rl}} = \min \left\{ \frac{b_i}{a_{il}} \mid a_{il} \geq 0 \right\}.$$

a_{rl} – разрешающий (направляющий) элемент, строка r – разрешающая.

Для перехода к следующей симплексной таблице (следующему опорному плану с меньшим значением целевой ф-ции) делается шаг модифицированного жорданова исключения с разрешающим эл-том a_{rl} .

Ес. в разрешающем столбце нет положительных коэф-тов, то целевая ф-ция неограничена снизу (при максимизации – неограничена сверху).

Шаг модифицированного жорданова исключения над симплексной таблицей:

1. На месте разрешающего элемента ставится 1 и делится на разрешающий элемент.
2. Остальные элементы разрешающего столбца меняют знак на противоположный и делятся на разрешающий элемент.
3. Остальные элементы разрешающей строки делятся на разрешающий элемент.
4. Все остальные элементы симплексной таблицы вычисляются по следующей ф-ле:

$$a_{ij} = \frac{a_{ij} \cdot a_{rl} - a_{rj} \cdot a_{il}}{a_{rl}} = a_{ij} - \frac{a_{rj} \cdot a_{il}}{a_{rl}}.$$

	$-x_4$	$-x_5$	1
x_1	1	-2	1
x_2	-2	1	2
x_3	3	1	3
$Q(\bar{x})$	-1	1	0

Разрешающий элемент, соответствующий замене базисной переменной x_2 на x_5

	$-x_4$	$-x_2$	1
x_1	-3	2	5
x_5	-2	1	2
x_3	5	-1	1
$Q(\bar{x})$	1	-1	-2

Разрешающий элемент, соответствующий замене базисной переменной x_3 на небазисную переменную x_4 .

	$-x_1$	$-x_2$	$-x_s$	$-x_n$	1
0	$a_{1,1}$	$a_{1,2}$	$a_{1,s}$	$a_{1,n}$	b_1
....
0	$a_{m,1}$	$a_{m,2}$	$a_{m,s}$	$a_{m,n}$	b_m
x_{m+1}	$a_{m+1,1}$	$a_{m+1,2}$	$a_{m+1,s}$	$a_{m+1,n}$	b_{m+1}
....
x_{m+p}	$a_{m+p,1}$	$a_{m+p,2}$	$a_{m+p,s}$	$a_{m+p,n}$	b_{m+p}
$Q(\bar{x})$	$-c_1$	$-c_2$	$-c_s$	$-c_n$	0

Правила выбора разрешающего элемента при поиске опорного плана:

1. При условии отсутствия “0-строк” (ограничений-равенств) и “свободных” переменных (т.е. переменных, на которые не наложено требование неотрицательности).

- Ес. в столбце свободных членов симплексной таблицы нет отрицательных элементов, то опорный план найден.
- Есть отрицательные элементы в столбце свободных членов, например $b_i < 0$. В такой строке ищем отрицательный коэффициент a_{il} , и этим самым определяем разрешающий столбец l . Ес. не найдем отрицательный a_{il} , то система ограничений несовместна (противоречива).
- В качестве разрешающей выбираем строку, которой соответствует минимальное отношение:

$$\frac{b_r}{a_{rl}} = \min_i \left\{ \frac{b_i}{a_{il}} \mid \frac{b_i}{a_{il}} > 0 \right\},$$

где r - номер разрешающей строки. Т.о., a_{rl} - разрешающий элемент.

- После того, как разрешающий элемент найден, делаем шаг модифицированного жорданова исключения с направляющим элементом a_{rl} и переходим к следующей симплексной таблице.

2. В случае присутствия ограничений-равенств и “свободных” переменных поступают следующим образом.

- Выбирают разрешающий элемент в “0-строке” и делают шаг модифицированного жорданова исключения, после чего вычеркивают этот разрешающий столбец. Данную по=>сть действий продолжают до тех пор, пока в симплексной таблице остается хотя бы одна “0-строка” (при этом таблица сокращается).
- Ес. же присутствуют и свободные переменные, то необх. данные переменные сделать базисными. И после того, как свободная переменная станет базисной, далее в процессе определения разрешающего элемента при поиске опорного и оптимального планов данная строка не учитывается (но преобразуется).

2.Алгоритмы глобального поиска.

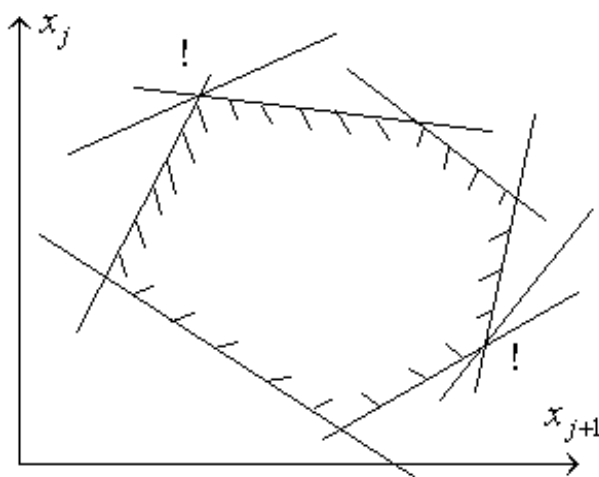
(см. 19-ый билет, 1-ый вопрос)

23 билет

1.Вырожденность в задачах линейного программирования.

Рассматривая симплекс-метод, мы предполагали, что задача линейного программирования явл-ся невырожденной, т.е. каждый опорный план содержит ровно m положительных компонент, где m – число ограничений в задаче. В вырожденном опорном плане число положительных компонент оказывается меньше числа ограничений: некоторые базисные переменные, соответствующие данному опорному плану, принимают нулевые значения.

Используя геометрическую интерпретацию для простейшего случая, когда $n - m = 2$ (число небазисных переменных равно 2), легко отличить вырожденную задачу от невырожденной. В вырожденной задаче в одной вершине многогранника условий пересекается более двух прямых, описываемых уравнениями вида $x_i = 0$. Это значит, что одна или



несколько сторон многоугольника условий стягиваются в точку.

Аналогично при $n - m = 3$ в вырожденной задаче в одной вершине пересекается более 3-х плоскостей $x_i = 0$.

В предположении о невырожденности задачи находилось только одно значение, соответствующее

минимуму $\theta = \min_i \left\{ \frac{b_i}{a_{il}} \mid \frac{b_i}{a_{il}} > 0 \right\}$, по которому определялся индекс

выводимого из базиса вектора условий (выводимой из числа базисных переменной).

В вырожденной задаче $\min_i \left\{ \frac{b_i}{a_{il}} \mid \frac{b_i}{a_{il}} > 0 \right\}$ может достигаться на нескольких

индексах сразу (для нескольких строк). В этом случае в нахожим опорном плане несколько базисных переменных будут нулевыми.

Ес. задача линейного программирования оказывается вырожденной, то при плохом выборе вектора условий, выводимого из базиса, может возникнуть бесконечное движение по базисам одного и того же опорного плана. Так называемое явление закливания. Хотя в практических задачах линейного программирования закливание явление крайне редкое, возможность его не исключена.

Один из приемов борьбы с вырожденностью состоит в преобразовании задачи путем “незначительного” изменения вектора правых частей системы ограничений на величины ε_i , т.о., чт. задача стала невырожденной и, в то же время, чт. это изменение не повлияло реально на оптимальный план задачи. Чаше реализуемые алгоритмы включают в себя некоторые простые правила, снижающие вероятность возникновения закливания или его преодоления. Пусть переменную x_j необх. сделать базисной. Рассмотрим мн-во индексов E_0 , состоящее из тех i , для которых

достигается $\theta_0 = \min_i \left\{ \frac{b_i}{a_{il}} \mid \frac{b_i}{a_{il}} > 0 \right\}$. Мн-во индексов i , для которых

выполняется данное условие, обозначим ч/з E_0 . Ес. E_0 состоит из одного элемента, то из базиса исключается вектор условий A_i (переменная x_i делается небазисной).

Ес. E_0 состоит более чем из одного элемента, то составляется мн-во E_1 ,

которое состоит из $i \in E_0$, на которых достигается $\theta_1 = \min_{i \in E_0} \left\{ \frac{a_{il}}{a_{il}} \right\}$. Ес. E_1

состоит из одного индекса k , то из базиса выводится переменная x_k . В противном случае составляется мн-во E_2 и т.д.

Практически правилом надо пользоваться, ес. закливание уже обнаружено.

2.Задача о максимальном потоке в транспортной сети.

(см. 9-ый билет, 2-ой вопрос)

24 билет

1. Двойственность задачи линейного программирования.

Рассм. задачу максимизации лин. ф-мы (1) и задачу минимизации (2):

$$\left. \begin{aligned} Q(\bar{x}) &= \bar{p}^T \cdot \bar{x} \rightarrow \max \\ A \cdot \bar{x} &\leq \bar{b}, \\ \bar{x} &\geq 0 \end{aligned} \right\} \quad (1) \qquad \left. \begin{aligned} W(\bar{u}) &= \bar{b}^T \cdot \bar{u} \rightarrow \min \\ A^T \cdot \bar{x} &\geq \bar{p}, \\ \bar{u} &\geq 0 \end{aligned} \right\} \quad (2)$$

Задача (2) наз-ся двойственной по отн. к прямой (1) (и наоборот!).

Пример: Предприятие выпускает три вида продукции. Каждая продукция требует обработки на трех различных типах установок. Ресурс времени каждого типа установок ограничен. Известна прибыль от единицы каждого вида продукции p_1, p_2, p_3 . Ес. количество выпускаемой продукции каждого вида x_1, x_2, x_3 , тогда необх. максимизировать прибыль

$$Q(\bar{x}) = p_1 \cdot x_1 + p_2 \cdot x_2 + p_3 \cdot x_3 \rightarrow \max$$

при ограничениях следующего вида:

$$a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 \leq b_1,$$

$$a_{21} \cdot x_1 + a_{22} \cdot x_2 + a_{23} \cdot x_3 \leq b_2,$$

$$a_{31} \cdot x_1 + a_{32} \cdot x_2 + a_{33} \cdot x_3 \leq b_3, \quad \bar{x} \geq \bar{0},$$

где b_1, b_2, b_3 – ресурсы времени установок первого, второго и третьего типов. Величины a_{ij} опр-ют кол-во ресурса времени установки i -го типа, кот-е необх. для выпуска одной единицы прод-и j -го вида. Двойств-я к ней задача будет иметь вид $W(\bar{u}) = b_1 \cdot u_1 + b_2 \cdot u_2 + b_3 \cdot u_3 \rightarrow \min$

$$\text{при ограничениях: } \begin{cases} a_{11} \cdot u_1 + a_{21} \cdot u_2 + a_{31} \cdot u_3 \geq p_1, \\ a_{12} \cdot u_1 + a_{22} \cdot u_2 + a_{32} \cdot u_3 \geq p_2, \\ a_{13} \cdot u_1 + a_{23} \cdot u_2 + a_{33} \cdot u_3 \geq p_3, \\ \bar{u} \geq \bar{0}. \end{cases}$$

Здесь u_1 – это оценка (цена), соответствующая одной единице ограниченного ресурса по первой установке. И она равна величине, на которую могла бы увеличиться суммарная прибыль, ес. бы количество этого ограниченного ресурса увеличилось на единицу, и ес. это увеличение было бы использовано оптимально. Иными словами, u_1 – это количество прибыли, недополученной из-за нехватки единицы ограниченного ресурса b_1 . Аналогичным образом м. интерпретировать смысл величин u_2 и u_3 .

Преобразования при решении прямой и двойственной задач. Пусть

имеются прямая и двойственная задачи следующего вида:

Прямая задача:

$$\begin{aligned} Q(\bar{x}) &= \bar{p}^T \cdot \bar{x} \rightarrow \max \\ A \cdot \bar{x} &\leq \bar{b} \\ \bar{x} &\geq \bar{0} \end{aligned}$$

Двойственная к ней задача:

$$\begin{aligned} W(\bar{u}) &= \bar{b}^T \cdot \bar{u} \rightarrow \min \\ A^T \cdot \bar{u} &\geq \bar{p} \\ \bar{u} &\geq \bar{0} \end{aligned}$$

Представим ограничения в виде:

$$\begin{aligned} \bar{y} &= -A \cdot \bar{x} + \bar{b} \geq \bar{0} \\ \bar{x} &\geq \bar{0} \end{aligned}$$

$$\begin{aligned} \bar{v} &= A^T \cdot \bar{u} - \bar{p} \geq 0 \\ \bar{u} &\geq \bar{0} \end{aligned}$$

Для ограничений прямой задачи симплексная таблица имеет вид:

	$-x_1$	\dots	$-x_S$	\dots	$-x_n$	1
$y_1 =$	a_{11}	\dots	a_{1S}	\dots	a_{1n}	b_1
\dots	\dots	\dots	\dots	\dots	\dots	\dots
$y_r =$	a_{r1}	\dots	a_{rS}	\dots	a_{rn}	b_r
\dots	\dots	\dots	\dots	\dots	\dots	\dots
$y_m =$	a_{m1}	\dots	a_{mS}	\dots	a_{mn}	b_m
$Q(\bar{x}) =$	$-p_1$	\dots	$-p_S$	\dots	$-p_n$	0

Пусть a_{rS} - разрешающий элемент. Сделаем шаг модифицированного жорданова исключения и получим таблицу:

	$-x_1$	\dots	$-y_r$	\dots	$-x_n$	1
$y_1 =$	b_{11}	\dots	$-a_{1S}$	\dots	b_{1n}	$b_{1,n+1}$
\dots	\dots	\dots	\dots	\dots	\dots	\dots
$x_S =$	a_{r1}	\dots	1	\dots	a_{rn}	b_r
\dots	\dots	\dots	\dots	\dots	\dots	\dots
$y_m =$	b_{m1}	\dots	$-a_{mS}$	\dots	b_{mn}	$b_{m,n+1}$

$Q(\bar{x})$ =	$b_{m+1,1}$	\dots	p_s	\dots	$b_{m+1,n}$	$b_{m+1,n+1}$
-------------------	-------------	---------	-------	---------	-------------	---------------

где $b_{ij} = a_{ij} \cdot a_{rs} - a_{rj} \cdot a_{is}$, и всю данную табл. следует раздел. еще на a_{rs} .

Симплексную табл. для двойств-й задачи запишем, развернув ее на 90° :

	v_1 =	\dots	v_s =	\dots	v_n =	$W =$
u_1	a_{11}	\dots	a_{1s}	\dots	a_{1n}	b_1
\dots	\dots	\dots	\dots	\dots	\dots	\dots
u_r	a_{r1}	\dots	a_{rs}	\dots	a_{rn}	b_r
\dots	\dots	\dots	\dots	\dots	\dots	\dots
u_m	a_{m1}	\dots	a_{ms}	\dots	a_{mn}	b_m
1	$-p_1$	\dots	$-p_s$	\dots	$-p_n$	0

Пусть a_{rs} - направляющий элемент. Сделаем шаг обыкновенного жорданова исключения (отличие от модифицированного состоит в том, что элементы в разрешающей строке меняют знаки, а в столбце знаки сохраняются; в остальном преобразование остается тем же):

	v_1 =	\dots	u_r =	\dots	$v_n =$	$W =$
u_1	b_{11}	\dots	$-a_{1s}$	\dots	b_{1n}	$b_{1,n+1}$
\dots	\dots	\dots	\dots	\dots	\dots	\dots
v_s	a_{r1}	\dots	1	\dots	a_{rn}	b_r
\dots	\dots	\dots	\dots	\dots	\dots	\dots
u_m	b_{m1}	\dots	$-a_{ms}$	\dots	b_{mn}	$b_{m,n+1}$
1	$b_{m+1,1}$	\dots	p_s	\dots	$b_{m+1,n}$	$b_{m+1,n+1}$

где $b_{ij} = a_{ij} \cdot a_{rs} - a_{rj} \cdot a_{is}$, и всю данную табл. также разделить еще на a_{rs} .

Замечание: Не следует забывать при преобразованиях, что в данном случае у нас таблица развернута.

Т.о., нетрудно заметить, что шаг модифицированного жорданова исключения над симплексной таблицей прямой задачи соответствует шагу обыкновенного жорданова исключения над симплексной таблицей двойственной задачи. Эти взаимно двойственные задачи можно совместить в одной симплексной таблице:

	v_1	\dots	v_s	\dots	v_n	$W =$
	$=$		$=$		$=$	
	$-x_1$		$-x_s$		$-x_n$	1
$u_1 \quad y_1 =$	a_{11}	\dots	a_{1s}	\dots	a_{1n}	b_1
\dots	\dots	\dots	\dots	\dots	\dots	\dots
u_r	a_{r1}	\dots	a_{rs}	\dots	a_{rn}	b_r
$y_r =$						
\dots	\dots	\dots	\dots	\dots	\dots	\dots
u_m	a_{m1}	\dots	a_{ms}	\dots	a_{mn}	b_m
$y_m =$						
1	$-p_1$	\dots	$-p_s$	\dots	$-p_n$	0
$Q(\bar{x}) =$						

Можно показать, что, решая основную задачу линейного программирования, решаем и двойственную к ней. И наоборот. Причем, $\max Q = \min W$.

2.Алгоритмы сопряженных градиентов.

(см. билет №8, 1-ый вопрос)