Введение в теорию сложности

1) Индивидуальная и массовая задачи, кодировка задачи, алгоритм решения массовой задачи, временная сложность алгоритма.

Методичка, стр. 4-8
Массовая задача Π:
- список свободных параметров;
- формулировка свойств, которым должно удовлетворять решение задачи.

Π есть множество индивидуальных задач $I \in \Pi$. **Индивидуальная задача** получается, если всем параметрам присвоить конкретные значения.

Пусть Σ — конечный алфавит, а Σ^* — множество слов в этом алфавите. Отображение e: $\Pi \to \Sigma^*$ называется кодировкой задачи Π.

Алгоритм A решает массовую задачу Π, если для любой индивидуальной задачи $I \in \Pi$:
- A применим к I, то есть останавливается за конечное число шагов
- A дает решение I

Кодировка задачи P — такое отображение $e : P \to \Sigma^*$, обладающее следующими свойствами:
- Возможность однозначно декодировать, то есть у двух различных ИЗ не может быть одинаковых кодировок.
- e,e^{-1} — полиномиально вычислимы
- Кодировка не избыточна, то есть для любой другой кодировки e_1, удовлетворяющей 1 и 2 условиям справедливо:

$$\exists p(.) : \forall I \in \Pi \quad |e(I)| < p(e_1(I))$$

Язык массовой задачи — это множество правильных слов, то есть слов, соответствующих ИЗ, имеющим положительный ответ(подразумевается задача распознавания): $L(\Pi, e) = e(Y(\Pi)) = \{s \in \Sigma^* | s = e(I), I \in Y(\Pi)\}$
Язык алгоритма — множество слов, принимаемых A, то есть таких, на которых алгоритм останавливается в состоянии q_Y, что соответсвует "да": $L(A) = \{\sigma \in \Sigma^* | A(\sigma) = q_Y\}$
Алгоритм A **решает** массовую задачу Π, с кодировкой e, если $L(e,\Pi) = L(A)$ и $\forall s \in \Sigma^*$ А останавливается

Число шагов алгоритма A для входа $s \in \Sigma^*$ — это $t_A(s)$.
Временная сложность $T_A(n) = max_{s \in \Sigma^*, |s|<n}\{t_A(s)\}$.

2) Задачи распознавания свойств. Классы P и NP.

Методичка, стр. 8-11
Задача распознавания свойств -- массовая задача, предполагающая ответ "да" или "нет", в качестве своего решения.
- $D(\Pi)$ -- множество всех возможных значений параметров массовой задачи.
- $Y(\Pi)$ -- множество всех индивидуальных задач, ответом на которые является "да".

Класс полиномиально разрешимых задач (P) -- это такие задачи, временная сложность алгоритма решения которых ограниченна полиномом:
- $\exists A$ такой, что A решает массовую задачу Π с кодировкой e
- $\exists p(\cdot)$ -- полином такой, что $T_A(n) < p(n) \quad , \forall n \in Z_+$

Примеры неполиномиальных задач:
- алгоритмически неразрешимые задачи: $\forall A \quad \exists I \in \Pi$ такая, что A не применим к I, например, $t_A(e(I)) = \infty$
  - 10-я проблема Гильберта: по данному многочлену g с целыми коэффициентами выяснить, имеет ли уравнение $g = 0$ целочисленное решение
- задачи, для которых длина записи выхода превышает любой наперед заданный полином от длины входа
  - найти все маршруты в задаче коммивояжёра
- ∀A, решающего Π с кодировкой e, ∀p(·) ∃I ∈ Π: $t_A(e(I)) > p(|e(I)|)$

Класс недетерминированно полиномиальных задач (NP) -- это такие задачи, для которых существует алгоритм решения на недерменированной машине Тьюринга:

- $\exists \hat{A}$ для НДМТ такой, что $\hat{A}$ решает массовую задачу Π с кодировкой e
- $\exists p(\cdot)$ -- полином такой, что $\hat{T}_{\hat{A}}(n) < p(n) \quad , \forall n \in Z_+$

3) Теорема об экспоненциальной временной оценке для задач из класса NP.

Методичка, стр. 11

Для любой $\Pi \in NP$ существует ДМТ A, решающая ее с не более чем экспоненциальной временной сложностью: $T_A(n) \leqslant 2^{p(n)}$.

4) Класс co-NP. Пример задачи, допускающей хорошую характеризацию. Доказательство утверждения о взаимоотношении классов NPC и co-NP.

Методичка, стр. 12-14

Дополнительная задача $\overline{\Pi}$ к массовой задаче Π -- задача, получаемая из Π путем введения альтернативного вопроса. То есть если в Π спрашиваем "верно ли x", то в $\overline{\Pi}$ спрашиваем "верно ли, что $\neg x$ "

- $D(\overline{\Pi}) = D(\Pi)$
- $Y(\overline{\Pi}) = D(\Pi) \setminus Y(\Pi)$

Класс co-P -- $\{\overline{\Pi} | \Pi \in P\}$

- co-P = P.

Класс co-NP -- $\{\overline{\Pi} | \Pi \in NP\}$.

- co-NP = NP пока не удалось ни доказать, ни опровергнуть, но это вряд ли верно.
- $P \in NP \cap \text{co-NP}$

Массовая задача Π **допускает хорошую характеризацию**, если $\Pi \in \text{NP} \cap \text{co-NP}$

- пример такой задачи -- это задача определения простоты числа.
- $P \subseteq \text{NP} \cap \text{co-NP}$

Массовая задача Π' с кодировкой e' **полиномиально сводится** к задаче Π с кодировкой e, если любая индивидуальная задача $I' \in \Pi'$ может быть сведена за полиномиальное от её длины время к некоторой задаче $I \in \Pi$ с сохранением ответа.

Массовая задача Π называется **NP-полной (универсальной)**, если

- принадлежит классу NP: $\Pi \in \text{NP}$
- любая задача из NP полиномиально сводится к Π: $\forall \Pi' \in \text{NP} \quad \Pi' \propto \Pi$

Класс **NPC (NP-complete)** -- множество всех NP-полных задач.

5) Критерий NP-полноты. Д-во NP-полноты задачи ЦЛН

Методичка, стр. 15

Критерий NP-полноты. Массовая задача Π **NP-полна** тогда и только тогда, когда она принадлежит классу **NP** и к ней полиномиально сводится какая-либо **NP-полная** задача.

6) Д-во NP-полноты задачи 3-выполнимость. NP-трудные задачи

Методичка, стр. 17-18

Класс **NP-трудных** задач содержит:

1. задачи распознавания свойств Π, для которых
   - не доказано, что $\Pi \in \text{NP}$
   - $\exists \Pi' \in \text{NPC} : \Pi' \propto \Pi$
2. задачи оптимизации, для которых соответствующие задачи распознавания свойств $\Pi \in \text{NPC}$
3. любые задачи, к которым сводятся по Тьюрингу хотя бы одна **NP-полная** задача

7) Взаимоотношение классов P, NP и NPC, NP и co-NP. Класс PSPACE

Легко показать, что $P \subseteq \mathrm{NP} \cap \mathrm{co\text{-}NP}$. Рабочая *гипотеза*, что $P = \mathrm{NP} \cap \mathrm{co\text{-}NP}$.

Если для некоторой NP-полной задачи Π дополнительная к ней задача $\overline{\Pi} \in \mathrm{NP}$, то NP = co-NP
Класс **PSPACE** массовых задач -- класс алгоритмов, требующих не более, чем полиномиальной памяти.
Гипотеза. $\mathrm{P} \subset \mathrm{PSPACE}$ (то есть, не факт, что вложение строгое, но скорее всего так). При этом NP-полные, NP-трудные, NP-эквивалентные задачи $\subset \mathrm{PSPACE} \setminus \mathrm{P}$

8) Псевдополиномиальные алгоритмы. Пример для задачи о рюкзаке

Псевдополиномиальный алгоритм - полиномиальный алгоритм, проявляющий экспоненциальный характер только при очень больших значениях числовых параметров.
Пусть $M(I)$ -- некоторая функция, задающая значение числового параметра индивидуальной задачи I. Если таких параметров несколько, в качестве $M(I)$ можно взять или максимальное, или среднее значение, а если задача вовсе не имеет числовых параметров (например, раскраска графа, шахматы и т.п.), то $M(I) = 0$. Алгоритм называется псевдополиномиальным, если он имеет оценку трудоемкости $T_{max}(I) = O(p(\,|\,I\,|\,, M(I)))$, где $p(\cdot,\cdot)$ -- некоторый полином от двух переменных.
en-wiki

9) Сильная NP-полнота. Теорема о связи сильной NP-полноты задачи с существованием псевдополиномиального алгоритма ее решения

Полиномиальное сужение массовой задачи Π -- множество таких индивидуальных задач I, числовые параметры которых не превосходят полинома от длины входа: $\Pi_{p(\cdot)} = \{I \in \Pi | M(I) \leqslant p(|I|)\}$
Массовая задача Π называется **сильно NP-полной**, если её полиномиальное сужение является NP-полным.
Примеры:

- задача выполнимости, задача 3-выполнимости -- совпадают со своими полиномиальными сужениями
- задача булевых линейных неравенств -- ВЫП сводится к её полиномиальноу сужению, где числовые параметры (правая часть неравенств) линейны.
- задача о целочисленном решении системы линейных уравнений -- , т.к. БЛН сводится к ней
- задача коммивояжёра (TSL) -- совпадает со своим сужением

Задача о рюкзаке -- слабо-NPC.
Теорема. Если NP не совпадает с P, то ни для какой сильно-NPC задачи не существует псевдополиномиального решения.

10)Определение ε-приближенного алгоритма и полностью полиномиальной приближенной схемы (ППП). Связь между существованием ППП и псевдополиномиальностью

Методичка, стр. 22-24

Задача дискретной оптимизации -- решение каждой индивидуальной задачи $I \in \Pi$ является произвольная реализация оптимума $\mathrm{Opt}_\Pi(I) = \max\limits_{z \in S_\Pi(I)} f_\Pi(I, z)$, где

- $S_\Pi(I)$ -- область допустимых значений дискретной переменной z
- f_Π -- целевая функция задачи оптимизации
- $\max$ вообще говоря вполне может быть заменён на $\min$

Алгоритм A называется **приближённым алгоритмом** решения массовой задачи Π, если для любой задачи $I \in \Pi$ он находит точку $z_A(I) \in S_\Pi(I)$, лежащую в области допустимых значений, принимаемую за приближённое решение.

Утверждение. Если $\mathrm{P} \neq \mathrm{NP}$, то ни для какой константы $C > 0$ не существует полиномиального приближённого алгоритма решения задачи о рюкзаке с оценкой $|\mathrm{Opt}(I) - A(I)| \leqslant C$.
Приближённый алгоритм A решения массовой задачи Π называется ε **-приближённым алгоритмом** решения задачи, если $\forall I \in \Pi: \left|\dfrac{\mathrm{Opt}_\Pi(I) - A(I)}{\mathrm{Opt}_\Pi(I)}\right| \leqslant \varepsilon$.

11)Теорема об отсутствии ПППС для задач оптимизации, соответствующих сильно NP-полным задачам распознавания

Методичка, стр. 24
Теорема Если для Π оптимизации

- соответствующая ей задача распознавания свойств является сильно NP-полной
- существует полином $\exists p'(\cdot) : \mathrm{Opt}_{\Pi}(I) < p'(M(I)) \quad \forall I \in \Pi$

то при условии, что $\mathrm{P} \neq \mathrm{NP}$ для Π не существует ППТС

Основы линейного программирования

12)Определение озЛП. Принцип граничных решений. Алгебраическая и битовая сложность ЛП. Результаты о сложности для задач, близких к ЛП

ЛП (линейное программирование) -- теория, приложения и методы решения системы линейных неравенств с конечным числом неизвестных : $Ax \leqslant b\ , \quad x = \{x_i\}, i = 1 \dots n$, существует ли $x \in \mathbb{R}^n$, удовлетворяющий данной системе линейных неравентсв

озЛП (основная задача линейного программирования) : найти такой вектор $x \in \mathbb{R}^n$ -- решение задачи линейного программирования $d^* = \max\limits_{x \in \mathbb{R}^n;\ Ax \leqslant b} \langle c, x \rangle$, максимизирующее линейную функцию $\langle c, x \rangle = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$

Утверждение (принцип граничных решений). Если озЛП имеет решение, то найдется такая подматрица A_I матрицы A, что любое решение системы уравнений $A_I x = b_I$ реализует максимум $\langle c, x \rangle$.
Алгебраическая сложность -- количество арифметических операций.
Битовая сложность -- количество операций с битами. Битовая сложность задач ЛП, ЛН полиномиальна. Вопрос о существовании алгебраически-полиномиального алгоритма для ЛП остается открытым.

13)Геометрическое описание симплекс-метода

(Копипаста из [ru.wiki], там-же есть хорошая иллюстрация.)
Симплекс-метод -- метод решения озЛП.

Каждое из линейных неравенств в $Ax \leqslant b$ ограничивает полупространство в соответствующем линейном пространстве. В результате все неравенства ограничивают некоторый многогранник (возможно, бесконечный), называемый также полиэдральным конусом. Уравнение $W(x) = c$, где $W(x)$ — максимизируемый (или минимизируемый) линейный функционал, порождает гиперплоскость $L(c)$. Зависимость от c порождает семейство параллельных гиперплоскостей. Тогда экстремальная задача приобретает следующую формулировку — требуется найти такое наибольшее c, что гиперплоскость $L(c)$ пересекает многогранник хотя бы в одной точке. Заметим, что пересечение оптимальной гиперплоскости и многогранника будет содержать хотя бы одну вершину. Принцип симплекс-метода состоит в том, что выбирается одна из вершин многогранника, после чего начинается движение по его ребрам от вершины к вершине в сторону увеличения значения функционала. Когда переход по ребру из текущей вершины в другую вершину с более высоким значением функционала невозможен, считается, что оптимальное значение с найдено.

14)Теорема о границах решений задач ЛП с целыми коэффициентами

Методичка, стр. 28-29
$\Delta(D) = \max |\det(D_1)|$, где D_1 — квадратная подматрица D.

Теорема (о границах решений). Если задача озЛП $d^* = \max\langle c, x \rangle, x \in \mathbb{R}^n, Ax \leqslant b$ размерности (n, m) с целыми коэффициентами разрешима, то у нее существует рациональное рашение x^* в шаре: $\|x^*\| \leqslant \sqrt{n}\Delta([A|b])$ и $d^* = \dfrac{t}{s}\ , \quad t, s \in Z, \quad |s| \leqslant \Delta(A)$

15)Теорема о мере несовместности систем линейных неравенств с целыми коэффициентами

Методичка, стр. 29

x^{ε} -- ε **-приближенное решение системы ЛН**, если

- в строчной записи: $\langle a_i, x^{\varepsilon} \rangle \leqslant b_i + \varepsilon \ , \quad \forall i \in [1, m]$
- в матричной записи: $Ax^{\varepsilon} \leqslant b + \varepsilon e$, где e -- вектор-столбец из единиц

Теорема. Если система линейных неравенств имеет ε_1 приближенное решение ($\varepsilon_1 = \dfrac{1}{(n+2)\Delta(A)}$), то эта система разрешима, то есть имеет точное решение.

16)Описание метода эллипсоидов

- *Методичка, стр. (30-32) 32-33*
- вики:Метод эллипсоидов

Решает задачу линейного программирования за полиномиальное число шагов.
Суть алгоритма в том, чтобы окружить данный многогранник эллипсоидом, а затем постепенно сжимать этот эллипсоид; оказывается, на каждом этапе объем эллипсоида уменьшается в константное число раз.

Лемма1. Если система $Ax \leqslant b$ совместна, то в шаре $E_0 = \|x\| \leqslant \sqrt{n}\Delta([A|b])$ найдется ее решение.
Таким образом получаем, что если система совместна, то эта лемма позволяет локализовать хотбы бы 1 из ее решений

Введем функцию невязки в точке x -- $t(x) = \max_i((Ax)_i - b_i)$. Точка $x^0 = \overline{0}$ -- это центр шара E_0. Если $t(x^0) \leqslant 0$, то x^0 -- решение. Если это не так, то возьмем s: $t(x) = \langle a_s, x^0 \rangle - b_s$, значит x^0 не удовлетворяет s-ому неравенству системы. Всякий вектор x, удовлетворяющий неравенству s, должен лежать в полупространстве $\leqslant \langle a_s, x^0 \rangle$. Пересечение этого полупространства с нашей сферой дают полуэлипсоид. Вокруг получившегося полуэлипсоида описываем новую сферу и повторяем алгоритм заново.

17)Теория двойственности ЛП

- *Методичка, стр. 35-36*
- http://www.mathelp.spb.ru/book1/lprog5.htm

Каждой задаче линейного программирования можно определенным образом сопоставить некоторую другую задачу (линейного программирования), называемую двойственной или сопряженной по отношению к исходной или прямой задаче.

Двойственной задачей к задаче линейного программирования $Ax \leqslant b$ на максимум $\langle c, x \rangle$ (в форме ОЗЛП можно записать: $\max\limits_{x \in \mathbb{R}^n:\ Ax \leqslant b} \langle c, x \rangle$) называется задача линейного программирования на минимум: $\min\limits_{\lambda \in \mathbb{R}^n:\ \lambda A = c,\ \lambda \geqslant \overline{0}} \langle \lambda, b \rangle$

Утверждение Двойственная задача к двойственной задаче совпадает с прямой задачей линейного программирования.

Теорема (двойственности ЛП). Задача ЛП разрешима тогда и только тогда, когда разрешима двойственная к ней. При этом в случае разрешимости оптимальные значения целевых функций совпадают: $\max\limits_{x \in \mathbb{R}^n:\ Ax \leqslant b} \langle c, x \rangle = \min\limits_{\lambda \in \mathbb{R}^n:\ \lambda A = c,\ \lambda \geqslant \overline{0}} \langle \lambda, b \rangle$

18)Сведение озЛП к однородной системе уравнений с огрничением x>0

Методичка, стр 36-37
Утверждение. Задача ЛП оптимизации эквивалентна решению системы линейных неравенств.
Утверждение. Задача ЛП оптимизации эквивалентна решению системы линейных уравнений в неотрицательных переменных.
Утверждение. Задача ЛП эквивалентна поиску неотрицательного ненулевого решения однородной системы линейных уравнений.

19)Идея метода Кармаркара

- *Методичка, стр 37-38*
- http://logic.pdmi.ras.ru/~yura/modern/02seminar.pdf

Метод Кармаркара.

1. На основании предыдущего утверждения (см. вопрос о сведении озЛП к однородной системе), есть возможность свести задачу ЛП $\max\limits_{x\in\mathbb{R}^n:\ Ax\leqslant b}\langle c,x\rangle$ к поиску решения СЛАУ $\hat{P}y=\hat{q},\ y\geqslant\overline{0}$, которая, в свою очередь, сводится к однородной СЛАУ: $Py=\overline{0},\ y\geqslant\overline{0}$
2. Введем **функцию Кармаркара**: $$k(x)=\frac{\left[(\langle p_1,x\rangle)^2+\ldots+(\langle p_K,x\rangle)^2\right]^{N/2}}{x_1\cdot x_2\cdot\ldots\cdot x_N}$$, где
   - N -- число столбцов в P
   - K -- число строк в P
   - $p_i,\ i\in[1,K]$ -- строки матрицы P
3. применяя теорему о мере несовместимости и алгоритм округления можно показать, что для решения достаточно найти такой $\hat{x}$, для которого $$k(\hat{x})\leqslant\frac{1}{3\left(\Delta(\hat{P})\right)^N}$$
4. при этом можно так же показать полиномиальный алгоритм поиска данного приближения, который в курсе не рассматривается.

20)Следствия систем линейных неравенств. Афинная лемма Фаркаша (без доказательства)

- *Методичка, стр. 34-35*
- http://imcs.dvgu.ru/lib/nurmi/finmath/node41.html

Система линейных неравенств $Ax\leqslant b$ называется **разрешимой**, если $\exists x:\quad Ax\leqslant b$

Линейное неравенство $\langle c,x\rangle\leqslant d$ является **следствием разрешимой системы ЛН** $Ax\leqslant b$, если для всех x, для которых выполняется сама система, выполняется и следствие: $\forall x: Ax\leqslant b\quad\Rightarrow\quad\langle c,x\rangle\leqslant d$

Афинная лемма Фаракша. Линейное неравентсво $\langle c,x\rangle\leqslant d$ является следствием разрешимой в вещественный переменных ЛН $Ax\leqslant b$, тогда и только тогда, когда существует $\lambda\in\mathbb{R}^m$:

- $c=\sum\limits_{i\in M}\lambda_i a_i$
- $d\geqslant\sum\limits_{i\in M}\lambda_i b_i$
- $\lambda_i\geqslant 0\ \ \forall i\in M$

21)Лемма Фаркаша о неразрешимости

Методичка, стр. 35

Лемма. Система линейных неравенств $Ax\leqslant b$ неразрешима тогда и только тогда, когда разрешима система:

- $\sum\limits_{i\in M}\lambda_i a_i=\overline{0}$ (нулевой вектор)
- $\sum\limits_{i\in M}\lambda_i b_i\leqslant -1$
- $\lambda_i\geqslant 0\ \ \forall i\in M$

Элементы математического программирования

22)Классификация задач математического программирования. Преимущества выпуклого случая

Методичка. стр 39-41

Задача математического программирования (ЗМП) -- по заданной $f(x)$ найти $\arg\min_{x \in X} f(x)$, то есть:

- найти $x^* \in X: \quad \forall x \in X \Rightarrow f(x^*) \leqslant f(x)$ -- решение
- $f^* = f(x^*)$ -- (оптимальное) значение целевой функции $f(x)$
- где X -- допустимое множество (множество ограничений)

Классификация проводится по *типу допустимого множества* X:

- *дискретные (комбинаторные)* -- множество X конечно или счётно
- *целочисленные* -- $X \equiv \mathbb{Z}^n$
- *булевы* -- $X \equiv \mathbb{B}^n$
- *непрерывные* -- $X \equiv \mathbb{R}^n$
- *бесконечномерные*
- *функциональные*

Задачи оптимизации бывают:

- *условные* -- $X \subset \mathbb{R}^n$
- *безусловные* -- $X \equiv \mathbb{R}^n$

Классификация по *свойствам целевой функции*: выпуклость, гладкость и т.п.

Классификация по результату:

- локальная оптимизация
- глобальная оптимизация

Выпуклое множество (вики) -- такое множество, которое содержит вместе с любыми двумя своими точками еще и отрезок, их соединяющий.

Функция f называется **выпуклой**, если её надграфик (множество точек над графиком: $\{(x, y): \ y \geqslant f(x) \ \forall x \in X\}$) является выпуклым множеством.

Утверждение. Любая точка локального минимума выпуклой функции является точкой её глобального минимума.

Преимущества выпуклых задач:

- применим метод эллипсоидов, причем сложность - полиномиальна
- для острых задач (целевая функция убывает в окрестности минимума не медленнее некоторой линейной функции) можно получить точное решение

23)Формула градиентного метода в задаче безусловной минимизации

Методичка. стр 41-42

Основная идея:

- берем некоторое начальное значение
- итеративно вычисляем градиент целевой функции
- двигаемся в обратном направлении
- и так постепенно приходим к (локальному) минимуму функции

Формула градиентного метода -- $x^{t+1} = x^t - \alpha_t \text{grad} f(x^t)$, где α_t -- шаговый множитель:

- пассивный способ: $\{\alpha_t\}$ выбирается заранее
- адаптивный способ: $\{\alpha_t\}$ выбирается в зависимости от реализующейся x_t

$$\alpha_t \in \arg\min_{\alpha > 0} f(x^t - \alpha \text{grad} f(x^t))$$

  - метод скорейшего спуска --
  - метод дробления (деления пополам) -- если $f(x^{t+1}) > f(x^t)$, то возвращаемся к шагу t с новым значением $\alpha_t = \alpha_t / 2$

24)Идея метода Ньютона

Методичка, стр. 43

Метод ньютона -- это фактически градиентный спуск с адаптивыным коэффициентом, который берется, как 2 производная целевой функции.

Реально можно вывести формулу Ньютона из разложения по Тейлору до 2 производной в окрестности точки минимума.

25)Формула метода Ньютона в задаче безусловной минимизации

Методичка. стр 43

Формула Ньютона -- $x^{t+1} = x^t - \frac{1}{f''(x^t)}\mathrm{grad} f(x^t)$, при этом начальное приближение должно находиться достаточно близко к искомой точке минимума.

Метод ньютона имеет квадратичную скорость сходимости: $\|x^{t+1} - x^*\| \leqslant \frac{1}{Q}(Q\|x^1 - x^*\|)^2$, где Q - некоторая константа

Ограничения:

- невырожденность матрицы 2 производных (гессиана)
- близость начального приближения к точке минимума ($\|x^1 - x^*\| < 1/Q$)

26)Идея метода штрафов

Методичка. стр 44

Смысл метода в том, чтобы свести задачу условной оптимизации к задаче безусловной оптимизации, то есть избавится от ограничения на область, в которой ищем минимум.

Для этого вводится так называемая функция штрафа, которая равна нулю в той области, в которой мы "условно оптимизируем" целевую функцию, а в остальных точках добавляет к значению целевой функции некоторое значение (собственно, штраф).

Пример. Пусть область задаётся следующим образом: $X = \{x | g(x) \leqslant 0\}$, где $g(x)$ -- некоторая функция. Тогда рассмотрим задачу безусловной минимизации целевой функции $f(x)$ со штрафом: $\min\limits_{x \in \mathbb{R}^n}\{f(x) + Cg(x)^p\}$, где C -- некоторая константа [??], а $p \geqslant 1$ -- параметр штрафа

Способы решения переборных задач

27)Методы глобальной минимизации

Методичка. стр. 52 (52-55)

28)Метод ветвей и границ для глобальной минимизации Липшицевых функций

Методичка. стр. 54

29)Метод ветвей и границ для ЦЛП. Различные стратегии метода

Методичка. стр. 57

30)Идея метода ветвей и границ. Пример для задачи БЛП

Методичка. стр. 59

31)Теорема оптимальности для разложимых функций

Методичка. стр 60

Опр. Функция f называется **разделяемой** на f_1 и f_2, если она представима в виде $f(x,y) = f_1(x, f_2(y))$.

Опр. Функция f называется **разложимой** на f_1 и f_2, если:

- она разделяема на f_1 и f_2
- f_1 монотонно не убывает по последнему аргументу

Теорема оптимальности для разложимых функций: $\min_{x,y}(f(x,y)) = \min_x(f_1(x, \min_y(f_2(y))))$.

Указанная теорема используется для уменьшения размерности оптимизационных задач и в методе ДП.

32)Применение метода динамического программирования для понижения размерности разложимой оптимизационной задачи

Методичка. стр. 62

33)Метод динамического программирования для БЛП с неотрицательными коэффициентами

Методичка. стр. 63-64

Неотсортировано

1. Полиномиальный алгоритм округления ε_1-приближенного решения системы линейных неравенств
2. Понятие о временной сложности алгоритмов
3. Понятие о недетерминированно-полиномиальных задачах
4. Оценка сложности метода эллипсоидов ε_2-приближенного решения озЛП