Методы оптимизациии. Теормин.

1. *Массовая задача.*Формально массовая задача **П** определяется:  
   1) общим списком всех параметров задачи (свободных параметров, значения которых не заданы),  
   2) формулировкой свойств, которым должен удовлетворять ответ (решение задачи).
2. *Индивидуальная задача.*Индивидуальная задача **I** ∈ **П** получается из **П**, если всем параметрам присвоить конкретные значения.
3. *Алгоритм решения задачи.*  
   Будем говорить, что алгоритм **А** решает массовую задачу **П**, если для ∀**I** ∈ **П** алгоритм **А** применим к **I** (т. е. останавливается за конечное число шагов) и для ∀**I** ∈ **П** алгоритм **А** дает решение задачи **I**.
4. *Полиномиальные алгоритмы решения массовой задачи.*Алгоритмы, решающие произвольную **I** ∈ **П** за время, ограниченное полиномом от “размера” **I**. Полиномиальные задачи - задачи, для которых существуют полиномиальные алгоритмы решения.
5. *Задачи распознавания свойств.*Массовые задачи, предполагающие ответ “да” или “нет” в качестве решения.
6. *Кодировка задачи.*Введем конечное множество - ***алфавит*** ∑ = {𝜎i}, а также множество ∑\* ***слов над алфавитом*** ∑  - произвольных конечных последовательностей, составленных из символов алфавита, возможно повторяющихся, 𝜎 = 𝜎i1𝜎i2...𝜎in, 𝜎ij ∈ ∑ ∀ij. Число n называется ***длиной слова*** 𝜎 и обозначается |𝜎|. ***Кодировкой задачи*** **П** назовем такое отображение *e*: **П** → ∑\*, ставящее в соответствие любой индивидуальной задаче **I** ∈ **П** ее код *e*(**I**) = 𝜎 ∈ ∑\* (слово из алфавита ∑\*), что  
       1) Возможно однозначное декодирование: ∀**I1** ≠ **I2** *e*(**I1**) ≠ e(**I2**)  
       2) *e, e-1* полиномиально вычислимы: существует алгоритм, реализующий e, e-1 и полином *p*(), для которого ∀**I** ∈ **П** время определения *e*(**I**), *e-1*(*e*(**I**)) не превосходит *p*(|*e*(**I**)|)   
       3) Кодировка неизбыточна: для любой другой кодировки *e*’, удовлетворяющей условиям 1) и 2), найдется полином *p*’, такой что ∀**I** ∈ **П** |*e*(**I**)| < *p*’(|*e*’(**I**)|).
7. *Временная сложность алгоритма.*Алгоритм **А** решает массовую задачу **П**, если L(**A**) = L(**П**, *e*) и  **А** останавливается. Обозначим *tA*(σ) – время работы над словом σ (число шагов).  
   **Временной сложностью** алгоритма **А** решения массовой задачи **П** назовем функцию
8. *Класс полиномиально разрешимых задач (P)***P** = {L(**П**, e) | }  
   Если для задачи **П** существует такая кодировка e, что , то будем называть **П** полиномиальной.
9. *Недетерминированная машина Тьюринга (НДМТ)*Определяется как набор обычных (детерминированных) машин Тьюринга A(S) с алфавитом Σ, где S пробегает все множество слов из Σ\*:   
   НДМТ останавливается, когда останавливается первая из ДМТ A(S), принимающая входное слово. Соответствующее конечное состояние – qY.  
   **Язык НДМТ** – множество слов, принимаемых хотя бы одной ДМТ.
10. *НДМТ решает массовую задачу* **П** *с кодировкой е*НДМТ решает массовую задачу **П** с кодировкой е, если L(П, e) = L(Â)
11. *Время работы НДМТ Â над словом σ*Минимальное из времен работы ДМТ A(S) над словом σ с учетом времени прочтения слова S
12. *Временная сложность НДМТ Â решения массовой задачи* **П**Функция
13. *Класс недерминированно полиномиальных задач (***NP***)*
14. *Теорема об экспоненциальной оценке временной сложности* **NP**
15. *Дополнительная массовая задача*Дополнительная к П массовая задача получается из П распознавания свойств заменой альтернативного вопроса, определяющего ответ в задаче его отрицанием.
16. *Классы co-P и co-NP***co-P** =   
    **co-NP** =   
    Утверждение: **co-P** = **P**.
17. *Задача, имеющая хорошую характеризацию***П** имеет хорошую характеризацию, если , **П** – распознавания свойств.
18. *Полиномиальная сводимость***П**’ распознавания свойств с кодировкой e’ полиномиально сводится к **П** с кодировкой e, если может быть сведена за полиномиальное от ее длины время к некоторой с сохранением ответа.
19. *Утверждения*
20. *NP-полная задача*Массовая задача П называется NP-полной (универсальной), если  
      
    Класс **NP**-полных задач – **NPC** (**NP**-complete).
21. *Задача о выполнимости (***ВЫП***)*Выяснить выполнимость КНФ (существует ли набор входных данных, на которых КНФ равна 1).
22. *Теорема Кука*
23. *Утверждения*Если , то P = NP  
    Если , то
24. *Критерий NP-полноты*Массовая задача П NP-полна тогда и только тогда, когда она принадлежит классу NP, и к ней полиномиально сводится какая-либо NP-полная задача.
25. *Задача* **ЦЛН**Задача о существовании целочисленного решения системы линейных неравенств с целыми коэффициентами. Принадлежит классу **NPC**.
26. *Задача* **БЛН**Задача о существовании булева решения системы линейных неравенств с целыми коэффициентами. Принадлежит классу **NPC**.
27. *Задача* **3-ВЫП**Частный случай **ВЫП**, когда функции от 3-х переменных.
28. *Утверждение*
29. *Утверждение*Если для некоторой NP-полной задачи П дополнительная к ней принадлежит классу NP, то NP = co-NP.
30. *Класс NP-трудных задач*Класс NP-трудных задач объемлет **NPC**. Он содержит:  
    1) **П** распознавания свойств, для которых доказано, что , но не доказано, что   
    2) **П** оптимизации, для которых соответствующие **П** распознавания свойтсв NP-полны  
    3) Остальные массовые задачи, к которым сводятся по Тьюринту какие-либо NP-полные задачи.
31. *NP-эквивалентные задачи***П**, для которых
32. *Класс PSPACE*Класс задач, для решения которых существуют алгоритмы, требующие не более чем полиномиальной памяти.
33. *Псевдополиномиальный алгоритм*Обозначим:  
    num(**I**) – максимальное по модулю число (или 0), фигурирующее при задании числовых параметров индивидуальной задачи **I**  
    |**I**| = |e(**I**)| – длину записи **I**.  
    Алгоритм А решения массовой задачи **П** называется **псевдополиномиальным**, если для некоторого полинома p() выполнено
34. *Полиномиальное сужение массовой задачи*Множество индивидуальных задач, числовые параметры которых не превосходят полинома от длины входа.
35. *Сильно NP-полная задача*Массовая задача П распознавания свойств называется сильно NP-полной, если ее полиномиальное сужение NP-полно.
36. *Теорема*Если P ≠ NP, то ни для какой сильно NP-полной задачи не существует псевдополиномиального алгоритма.
37. *Задачи дискретной (комбинаторной) оптимизации*Для оптимизационной постановки задачи П решением каждой **I** ∈ **П** является произвольная оптимизация SП – область допустимых значений дискретной (целочисленной) переменной z.  
     – целевая функция задачи **I**.
38. *Приближенный алгоритм решения*Алгоритм А называется **приближенным алгоритмом** решения массовой задачи П оптимизации, если ∀**I** ∈ **П** он находит некоторую точку из допустимой области  
     zA(**I**) ∈ S**П**(**I**), принимаемую за приближенное решение. Значение f**П**(**I**, zA(**I**)) называется приближенным значением оптимума и обозначается A(**I**).
39. *Утверждение о погрешности*Если P ≠ NP, то ни для какой константы *C* > 0 не существует полиномиального приближенного алгоритма А решения задачи о рюкзаке ЗР с оценкой
40. *ε-приближенный алгоритм*Приближенный алгоритм А решения массовой задачи П называется ε-приближенным для некоторого ε > 0, если
41. *Теорема*Пусть для задачи П оптимизации  
    1) Существует псевдополиномиальный алгоритм ее решения  
    2) для некоторых полиномов p1 и p2  
    3) ∀σ = e(I), I ∈ П, параметры σS, задающие ограничения, и σf, задающие целевую функцию, не пересекающиеся и ∀z ∈ SП(σ) функция цели fП(σ, z) линейно зависит от параметров σf. Тогда  
    *ε-*приближенный алгоритм Аε решения **П** с временной сложностью
42. *Полностью полиномиальная приближенная схема (ПППС)*Набор алгоритмов из теоремы в пункте 41.
43. *Теорема*Если для П оптимизации соответствующая ей П распознавания свойств является сильно NP-полной, и , то, при условии что P ≠ NP, для П не существует ПППС.
44. *Утверждение*Если P ≠ NP, то ни для какого ε > 0 не существует полиномиального ε-приближенного алгоритма решения оптимизационной постановки задачи коммивояжера.
45. *Основная задача линейного программирования (озЛП)*Состоит в нахождении такого решения системы ЛН , которое максимизирует целевую функцию.
46. *Каноническая задача ЛП*
47. *Принцип граничных решений*Если задача имеет решение, то найдется такая подматрица AI матрицы А, что любое решение системы уравнений AIx = bI реализует максимум в
48. *Симплекс-метод*Метод направленного перебора смежных вершин в направлении возрастания целевой функции.
49. *Функции алгебраической и битовой сложности***Функция алгебраической сложности**: оценивает число арифметических операций в зависимости от размерности, не учитывает длину кода элементов.  
    **Функция битовой сложности**: оценивает число арифметических операций с битами (или конечными порциями – по размеру машинного регистра) цифровой записи параметров задачи в зависимости от длины входного слова.
50. *Сильнополиномиальный метод*Метод, имеющий полиномиальную алгебраическую сложность.
51. *Теорема о границах решений*Для любой целочисленной матрицы D введем параметр  
      
    [A|b] – матрица, полученная приписыванием справа к А столбца b.  
    Теорема  
    Если озЛП размерности (n, m) с целыми коэффициентами разрешима, то у нее существует рациональное решение x\* в шаре и значением озЛП  
     d\* = <c, x\*> является рациональное число t/s со знаменателем, ограниченным величиной Δ(А).
52. *ε-приближенное решение системы ЛН*Точка xε называется ε-приближенным решением системы неравенств, если  
     , где аi – i-ая строка матрицы А.
53. *Теорема о мере несоместности*Если ЛН имеет ε1-приближенное решение для , то эта система разрешима, то есть имеет точное решение.
54. *ε-приближенное решение озЛП*Точка называется ε-приближенным решением озЛП, если она является ε-приближенным решением системы ЛН и реализует максимум с ε-точностью:
55. *Теорема о мере несоместности\**Если озЛП имеет ε2-приближенное решение для , то эта задача имеет точное решение.
56. *Следствие разрешимой системы ЛН*Линейное неравенство <c, x> ≤ d является следствием разрешимой системы ЛН, если для ∀x, удовлетворяющего системе, выполнено <c, x> ≤ d.
57. *Лемма Фаркаша (аффинная)*Линейное неравенство <c, x> ≤ d является следствием разрешимой в вещественных переменных системы ЛН тогда т только тогда, когда существует вектор
58. *Лемма Фаркаша о неразрешимости*Система ЛН Ax ≤ b неразрешима тогда и только тогда, когда разрешима система
59. *Двойственная задача*Двойственной к задаче ЛП на максимум с ограничениями неравенствами в форме озЛП называется следующая задача ЛП с ограничениями в канонической форме:  
     или кратко
60. *Теорема о двойственности ЛП*Задача ЛП разрешима тогда и только тогда, когда разрешима двойственная к ней. В случае разрешимости оптимальные значения целевых функций совпадают.
61. *Утверждение*Задача ЛП оптимизации эквивалентна решению системы ЛН.
62. *Утверждение*Задача ЛП эквивалентна решению системы линейных уравнений в неотрицательных переменных.
63. *Утверждение*Задача ЛП эквивалентна поиску неотрицательного ненулевого решения однородной системы линейных уравнений.
64. *Метод Крамакара*
65. *Классификация задач МП*ЗМП – задачи с вещественными переменными.  
     1) Условные задачи   
     2) Безусловные   
    По свойствам целевой функции ЗПМ делятся на  
     1) Задачи ЛП  
     2) Задачи выпуклого программирования  
     3) Гладкие и негладкие и др.  
    С точки зрения численных методов  
     Локальная и глобальная оптимизация
66. *Точка локального минимума в ЗМП* называется точкой локалького минимума в ЗМП, если
67. *Утверждение*
68. *Выпуклая функция*Функция f(x) называется выпуклой на X, если ее надграфик  
     – выпуклое множество. Функция называется выпуклой, если она выпукла на всей области определения.  
    Множество называется выпуклым, если для любых двух своих точек оно содержит отрезок, их соединяющий.
69. *Утверждение*Любая точка локального минимума выпуклой функции является точкой ее глобального минимума.
70. *Преимущества выпуклого случая* 1) Задача поиска ε-приближенного решения задачи выпуклого программирования полиномиально разрешима  
     2) Для острых задач – когда функция цели убывает в окрестности минимума не медленнее некоторой линейной функции, можно найти точное решение.
71. *Направление убывания функции*Вектор называется вектором убывания f в точке x, если для ∀ достаточно малых
72. *Утверждение*Пусть f дифференцируема в X. Тогда, если , то h – направление убывания f в точке x. Если h – направление убывания f в точке x, то
73. *Градиентный метод оптимизации*Выбор :  
     1) Пассивные способы – выбираются заранее  
     2) Адаптивные способы – зависит от реализующейся итерации (метод скорейшего спуска, метод дробления шага, правило Армихо)
74. *Метод проекции градиента*
75. *Дифференцируемая по Адамару функция*Функция называется дифференцируемой по Адамару в точке , если существует вектор , такой что выполнено
76. *Контингентный конус*Контингентным конусом ко множеству X в точке x называется множество векторов
77. *Общий вид необходимых условий локального минимума*Пусть f дифференцируема по Адамару, – точка лоального минимума f. Тогда
78. *Регулярное множество  
     –* множество активных ограничений.  
    Множество X для ограничений неравенств называется **регулярным** в точке x ∈ X, если
79. *Необходимые условия локального минимума с ограничениями-неравенствами*Пусть дифференцируемы по Адамару, – точка локального минимума *f* и X регулярно в x0. Тогда
80. *Принцип оптимальности Лагранжа*В предположениях теоремы из пункта 79 существует неотрицательный вектор множителей Лагранжа , такой что для x0 выполнены условия оптимальности
81. *Теорема Куна-Таккера*Если функции выпуклы и множество X регулярно в любой точке, то x\* – точка оптимума в этой задаче тогда и только тогда, когда в ней выполнены условия оптимальности для
82. *Вполне унимодулярная матрица*Матрица называется унимодулярной, если определительн любой ее невыожденной квадратной подматрицы равен по модулю 1.