Федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (национальный исследовательский университет)»

УТВЕРЖДЕНО
Директор высшей школы программной инженерии
А.В. Малеев

Рабочая программа дисциплины (модуля)

по дисциплине:	Дискретная оптимизация
по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии МФТИ - Яндекс
курс:	3
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 6 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:
лекции: 30 час.
семинары: 30 час.
лабораторные занятия: 0 час.

Самостоятельная работа: 48 час.

Всего часов: 108, всего зач. ед.: 3

Количество контрольных работ, заданий: 1

Программу составил: А.В. Созыкин, канд. техн. наук, доцент

Программа обсуждена на заседании высшей школы программной инженерии МФТИ - Яндекс 28.04.2023

Аннотация

В большинство курсов алгоритмов и структур данных входят такие классические темы, как поиск кратчайших путей в графах, построение кратчайших остовных деревьев, нахождение максимальных потоков. Всё это — примеры задач дискретной оптимизации, а общая схема вопросов дискретной оптимизации такова: построить дискретный объект, на котором достигается оптимальное значение некоторой «метрики качества». В реальной жизни редко можно встретить изолированные постановки вроде «найти путь между двумя вершинами графа». Всерьёз важные для практики задачи часто весьма трудны для точного решения, а лишь стандартными алгоритмами из вводных курсов, и даже их комбинациями, тут не обойтись. Мы рассмотрим в курсе общие подходы к решению задач дискретной оптимизации, все этапы — от построения сбалансированной математической модели до написания эффективных алгоритмов её обсчёта.

1. Цели и задачи

Цель дисциплины

- изучение классических и современных методов оптимизации. Рассмотрение примеров их использования в прикладных задачах физики, математики и информатики.

Задачи дисциплины

- изучение математических основ современной комбинаторики;
- приобретение слушателями теоретических знаний в области комбинаторного анализа задач, возникающих на практике;
- освоение аналитического и алгебраического аппарата дискретной математики и получение навыков работы с основными дискретными структурами.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
ОПК-1 Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.3 Знает методы тестирования программного кода на ошибки и способен проводить тестирование на различных уровнях (модульное, интеграционное, системное)
ОПК-8 Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий	ОПК-8.1 Понимает принципы, по которым работают базы данных, и умеет создавать структуру данных, оптимизированную для эффективного хранения и обработки информации
	ОПК-8.2 Умеет примененять технологии машинного обучения в различных прикладных областях
ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.2 Умеет выбирать языки программирования для написания программного кода с учетом технического задания

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- фундаментальные понятия, законы, теории части дискретной математики;
- современные проблемы соответствующих разделов дискретной математики;
- понятия, аксиомы, методы доказательств и доказательства основных теорем в разделах, входящих в базовую часть цикла;
- основные свойства соответствующих математических объектов;
- аналитические и численные подходы и методы для решения типовых прикладных задач дискретной математики.

уметь:

- понять поставленную задачу;
- использовать свои знания для решения фундаментальных и прикладных задач;
- оценивать корректность постановок задач;
- строго доказывать или опровергать утверждение;
- самостоятельно находить алгоритмы решения задач, в том числе и нестандартных, и проводить их анализ;
- самостоятельно видеть следствия полученных результатов;
- точно представить математические знания в области в устной и письменной форме.

владеть:

- навыками освоения большого объема информации и решения задач (в том числе, сложных);
- навыками самостоятельной работы и освоения новых дисциплин;
- культурой постановки, анализа и решения математических и прикладных задач, требующих для своего решения использования математических подходов и методов;
- предметным языком дискретной математики и навыками грамотного описания решения задач и представления полученных результатов.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Алгоритмы Прима и Борувки для решения задачи MST	4	4		6
2	Двойственность в линейном программировании	2	2		6
3	Дискретная линейная задача о подмножестве (DLS problem)	5	5		6
4	Задача построения паросочетания максимальной мощности в произвольном графе	4	4		6
5	Метод ветвей и границ	4	4		8
6	Модификации алгоритма Дейкстры	6	6		8
7	Отличительные особенности задач дискретной оптимизации	5	5		8
Итого часов		30	30		48
Подготовка к экзамену		0 час.			
Общая трудоёмкость		108 час., 3 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 6 (Весенний)

1. Алгоритмы Прима и Борувки для решения задачи MST

Напоминание основных понятий из линейного программирования. Задача в стандартной и канонической формах. Переход от неравенств к равенствам и обратно. Геометрия задачи: симплекс-алгоритм как локальный поиск по вершинам многогранника.

2. Двойственность в линейном программировании

Постановки задачи TSP в терминах ЦЛП. Условия Миллера—Таккера—Землина (полиномиальное количество неравенств в задаче TSP). Замечание «о некатастрофичности экспоненциального числа ограничений в задачах ЛП».

3. Дискретная линейная задача о подмножестве (DLS problem)

Задачи TSP и MST как частные случаи DLS-задач минимизации; переход к максимизации. Наследственные системы. Базы и циклы. Ранг и нижний ранг множества, ранговый разброс. Матроиды: эквивалентные определения, примеры. Оценка качества работы жадного алгоритма на наследственной системе через её ранговый разброс. Следствие о корректности жадного алгоритма построения кратчайшего остовного дерева. Оценка рангового разброса через ограничение на число циклов. Субмодулярность ранговой функции матроида. Перечесение матроидов. Оценка числа циклов для наследственной системы через число матроидов в пересечении.

4. Задача построения паросочетания максимальной мощности в произвольном графе

Увеличивающие пути (утверждение о том, что паросочетание немаксимально ⇔ есть увеличивающий путь). Проблема с поиском увеличивающих путей при отсутствии двудольности: цветки. Утверждения о сжатии цветков. Алгоритм Эдмондса.

5. Метод ветвей и границ

Задачи исчерпывающего перебора сложных дискретных объектов. Подход Рида: упорядоченное перечисление. Метод обращения локального поиска Ависа—Фукуды.

6. Модификации алгоритма Дейкстры

Два алгоритма: постепенная минимизация стоимости потока при неизменной величине; приращение величины за счёт минимально возможного приращения стоимости.

7. Отличительные особенности задач дискретной оптимизации

Обзор постановок классических задач дискретной оптимизации: покрытие множествами, вершинное покрытие, кратчайший путь, минимальное остовное дерево, задачи о паросочетании, задача о назначениях, задачи теории расписаний, задачи упаковки (bin packing, рюкзак), задачи о потоках (поток наибольшей величины, поток наименьшей стоимости, мультипродуктовые потоки), транспортная задача (задача Хичкока), задача коммивояжёра.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор, звуковая система).

6.Перечень рекомендуемой литературы

Основная литература

1. Дискретная математика. Теория и практика решения задач по информатике, учебное пособие / С. М. Окулов. — Москва, Лаборатория знаний, 2020.— URL: http://books.mipt.ru/book/301446 (дата обращения: 18.02.2021). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

-

Дополнительная литература

-

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

- https://developers.google.com/optimization?hl=ru

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

На лекционных занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

В процессе самостоятельной работы обучающихся возможно использование таких программных средств, как Mathcad, MATLAB, Maple и др.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

1. Рекомендуется успешно сдавать контрольные работы, так как это упрощает итоговую аттестацию по предмету.
2. Для подготовки к итоговой аттестации по предмету лучше всего пользоваться материалами лекций.

ПРИЛОЖЕНИЕ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии МФТИ - Яндекс высшая школа программной инженерии
курс:	3
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 6 (весенний) - Дифференцированный зачет

Разработчик: А.В. Созыкин, канд. техн. наук, доцент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
ОПК-1 Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.3 Знает методы тестирования программного кода на ошибки и способен проводить тестирование на различных уровнях (модульное, интеграционное, системное)
ОПК-8 Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий	ОПК-8.1 Понимает принципы, по которым работают базы данных, и умеет создавать структуру данных, оптимизированную для эффективного хранения и обработки информации
	ОПК-8.2 Умеет применентять технологии машинного обучения в различных прикладных областях
ПК-3 Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение	ПК-3.2 Умеет выбирать языки программирования для написания программного кода с учетом технического задания

2. Показатели оценивания компетенций

В результате изучения дисциплины «Дискретная оптимизация» обучающийся должен:

знать:

- фундаментальные понятия, законы, теории части дискретной математики;
- современные проблемы соответствующих разделов дискретной математики;
- понятия, аксиомы, методы доказательств и доказательства основных теорем в разделах, входящих в базовую часть цикла;
- основные свойства соответствующих математических объектов;
- аналитические и численные подходы и методы для решения типовых прикладных задач дискретной математики.

уметь:

- понять поставленную задачу;
- использовать свои знания для решения фундаментальных и прикладных задач;
- оценивать корректность постановок задач;
- строго доказывать или опровергать утверждение;
- самостоятельно находить алгоритмы решения задач, в том числе и нестандартных, и проводить их анализ;
- самостоятельно видеть следствия полученных результатов;
- точно представить математические знания в области в устной и письменной форме.

владеть:

- навыками освоения большого объема информации и решения задач (в том числе, сложных);
- навыками самостоятельной работы и освоения новых дисциплин;
- культурой постановки, анализа и решения математических и прикладных задач, требующих для своего решения использования математических подходов и методов;
- предметным языком дискретной математики и навыками грамотного описания решения задач и представления полученных результатов.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

Текущий контроль

Даны: список чисел [w_1, \dots , w_k], w_i \in [0, 1][w 1 ,…,w k],w i∈[0,1] — объёмы предметов, натуральное число n_{\text{bins}}n bins — количество контейнеров. Вместимость каждого контейнера полагается равной единице. Предполагается, что функция solve_bp_decision(weights, n_bins) решает «распознавательный» (decision) вариант задачи bin packing. На вход ей подаётся список предметов и число контейнеров. Функция возвращает булевский ответ на вопрос «можно ли заданные веса распределить по не более чем n_bins контейнерам? Напишите содержимое функции solve_bp_evaluation, которая возвращает оптимальное число контейнеров (решает evaluation-вариант задачи bin packing). Функция solve_bp_evaluation должна решать свою задачу не напрямую, а посредством вызова функции solve_bp_decision не более чем полиномиальное по размеру задачи количество раз. Не допускается использование рекурсивных вызовов функции solve_bp_evaluation внутри неё самой. Затем напишите содержимое функции solve_bp_search, которая возвращает список номеров контейнеров, которые при каком-нибудь оптимальном распределении присваиваются весам из списка weights (контейнеры нумеруются с единицы!). Функция solve_bp_search должна решать свою задачу не напрямую, а посредством вызова функции solve_bp_evaluation не более чем полиномиальное по размеру задачи количество раз. Не допускается использование рекурсивных вызовов функции solve_bp_search внутри неё самой.

Реализуйте приближённый алгоритм для задачи о покрытии, основанный на следующей схеме:

1) Кодируем задачу о покрытии как задачу линейного программирования, снимая ограничения на целочисленность переменных.

2) Запускаем библиотечный солвер для точного (!) решения задачи линейного программирования.

3) Используем решение задачи ЛП, выданное солвером, для формирования «настоящих индикаторных» значений переменных с помощью вероятностного округления. Приветствуется изобретение собственных эвристик-надстроек над схемой вероятностного округления. Например, как только уже округлённых к единице переменных хватает, чтобы покрыть матрицу, можно остановиться, не дожидаясь конца раунда округления. Помогает ли в этом случае упорядочение переменных (ещё не округлённых) по убыванию или по возрастанию? Кроме того, можно провести несколько десятков раундов вероятностного округления, а затем выбрать из них наименьшее по весу получившееся покрытие. Отдельно опишите используемую Вами стратегию.

Вам предлагается реализовать переборный алгоритм, который находит гарантированно точно максимальную стоимость рюкзака. В ходе перебора следует выполнять отсечение, используя верхнюю и нижнюю оценку стоимости рюкзака. Верхняя оценка — на основе (точного) решения нецелочисленной задачи ЛП, а нижняя — на основе жадного алгоритма решения задачи о рюкзаке. Программа должна печатать в стандартный поток вывода единственное число: стоимость оптимального рюкзака.

Запрограммируйте алгоритм 2-аппроксимации дерева Штейнера (который мы разбирали на занятии), работающий для произвольного, возможно, не метрического, графа:

1) Строим метрическое замыкание графа.

2) Строим минимальное остовное дерево только на обязательных вершинах.

3) Возвращаемся к исходному графу, получая, возможно, недревесное подмножество рёбер. 4) Удаляем (например, жадно), лишние рёбра, чтобы получить полноценное дерево Штейнера. Данные предоставляются в формате STP. Выход программы должен содержать несколько строк, в каждой из которых через пробел записаны id вершин, образующих ребро в дереве Штейнера.

Запрограммируйте генетический алгоритм для общей задачи о покрытии, сфомулированной в терминологии Set Cover. В этой задаче предполагается, что есть набор элементов и набор множеств. Требуется покрыть все элементы, взяв несколько множеств, так, чтобы минимизировать сумму весов взятых множеств. Для задания [части] первоначальной популяции используйте, например жадный подход. Алгоритм должен работать так, чтобы останавливаться либо по достижении локального оптимума, либо по приближении лимита по времени.

Короткие вопросы:

Сколько гамильтоновых циклов лежит в 2-окрестности фиксированного гамильтонова цикла в евклидовой задаче коммивояжёра для n точек? Укажите верный порядок этой величины.

Считая, что функции мутации и скрещивания вычисляются за один такт времени, сколько (по порядку) времени уйдёт на N шагов генетического алгоритма, работающего с популяцией размера m?

Считая, что в алгоритме локального поиска Кернигана—Лина в качестве “короткого шага” используются обычные для задачи TSP 2-окрестности, и что алгоритм делает за один “длинный шаг” максимум 5 коротких шагов. Сколько по порядку времени уйдёт на N шагов такого поиска Кернигана—Лина в евклидовой задаче коммивояжёра для m точек?

Какой из перечисленных на лекции алгоритмов для задачи Bin Packing обладает лучшим теоретически обоснованным показателем аппроксимации?

Кримеры курсовых работ:

Жадный алгоритм;

Алгоритмы Прима и Борувки;

Задачи о распределении дискретного однородного ресурса;

Понятие о сглаженном анализе алгоритмов;

Комбинаторный алгоритм для задач;

Задачи о покрытии;

«Биологические» метаэвристики.

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

1. Отличительные особенности задач дискретной оптимизации. Обзор постановок классических задач дискретной оптимизации: покрытие множествами, вершинное покрытие, кратчайший путь, минимальное остовное дерево, задачи о паросочетании, задача о назначениях, задачи теории расписаний, задачи упаковки (bin packing, рюкзак), задачи о потоках (поток наибольшей величины, поток наименьшей стоимости, мультипродуктовые потоки), транспортная задача (задача Хичкока), задача коммивояжёра.
2. Локальный поиск как широкий общий подход к решению задач дискретной оптимизации. Системы окрестностей. Пример системы окрестностей в задаче TSP: компромисс между силой окрестности и размером. Пример, в котором 2-окрестность не позволяет достичь глобального оптимума. Эвристика Кернигана—Лина: локальный поиск переменной глубины. Надстройки над локальным поиском: имитация отжига и табу-поиск.
3. Несуществование полиномиально обозримой точной системы окрестностей в задаче TSP (в предположении P≠NP). Локальные жадные эвристики в задаче TSP, не укладывающиеся явно в парадигму локального поиска (не переходящие от цикла к циклу, а строящие цикл с нуля). Показатели качества работы эвристических (приближённых) алгоритмов: appproximation ratio и domination number. Алгоритм ближайшего соседа (nearest neighbor): идея, теорема о том, что approximation ratio оценивается сверху O (log #вершин).
4. Дискретная линейная задача о подмножестве (DLS problem). Задачи TSP и MST как частные случаи DLS-задач минимизации; переход к максимизации. Наследственные системы. Базы и циклы. Ранг и нижний ранг множества, ранговый разброс. Матроиды: эквивалентные определения, примеры. Оценка качества работы жадного алгоритма на наследственной системе через её ранговый разброс. Следствие о корректности жадного алгоритма построения кратчайшего остовного дерева. Оценка рангового разброса через ограничение на число циклов. Субмодулярность ранговой функции матроида. Перечесение матроидов. Оценка числа циклов для наследственной системы через число матроидов в пересечении. Вероятность единственности решения задачи DLS при случайном выборе весов: лемма об изолировании.
5. Алгоритмы Прима и Борувки для решения задачи MST: примеры, реализация (без использования куч). Алгоритм Прима с использованием фибоначчиевых куч.

6. Задачи о распределении дискретного однородного ресурса: задача дискретного максимина, максимизация суммы вогнутых функций. Критерии оптимальности (принцип уравнивания Гермейера, критерий Гросса). Алгоритм «двойного бинарного поиска». Оптимизация произведения при фиксированной сумме.

7. Напоминание основных понятий из линейного программирования. Задача в стандартной и канонической формах. Переход от неравенств к равенствам и обратно. Геометрия задачи: симплекс-алгоритм как локальный поиск по вершинам многогранника.

8. Пример многогранника, на котором при некоторых условиях симплекс-алгоритму может потребоваться экспоненциально много шагов: теорема Кли—Минти. Верхняя оценка на число шагов «везучего симплекс-метода»: теорема Калаи—Клейтмана о диаметре графа многогранника.

9. Понятие о сглаженном анализе алгоритмов (smoothed analysis): среднее между анализом на случайных входах и анализом худшего случая. Теорема Спилмана—Тенга о симплекс-методе.

10. Двойственность в линейном программировании: решение двойственной задачи как сертификат оптимальности решения прямой задачи. Исключение Фурье—Моцкина. Лемма Фаркаша: существование сертификата неразрешимости системы линейных неравенств. Вывод теоремы о сильной двойственности из леммы Фаркаша.

11. Постановки задачи TSP в терминах ЦЛП. Условия Миллера—Таккера—Землина (полиномиальное количество неравенств в задаче TSP). Замечание «о некатастрофичности экспоненциального числа ограничений в задачах ЛП».

12. Простой «комбинаторный» алгоритм для задачи о вершинном покрытии (ВП) с approximation ratio = 2. Постановка задачи о взвешенном вершинном покрытии (ВВП) в терминах целочисленного линейного программирования. Алгоритм решения задачи ВВП вида «решаем задачу ЛП → округляем»; утверждение о том, что достигается approximation ratio = 2. Формулировка двойственной задачи к задаче ВВП: потенциалы на рёбрах. «Комбинаторный» (без использования ЛП) алгоритм решения задачи ВВП на основе двойственности; доказательство того, что для этого алгоритма approximation ratio = 2.

13. Общая задача о покрытии (эквивалентная задаче о покрытии множеств). Формулировка в терминах матриц. Постановка в виде ЦЛП, формулировка двойственной задачи. Теорема о том, что размер/вес жадного покрытия не больше чем в ln k раз превышает размер/вес оптимального (где k — максимальное число единиц в строке). Достижимость (по порядку) этой оценки. Оценка веса жадного покрытия через вес оптимального при ограниченных весах отдельных строк.

14. Задача построения паросочетания максимальной мощности в произвольном графе. Увеличивающие пути (утверждение о том, что паросочетание немаксимально ⇔ есть увеличивающий путь). Проблема с поиском увеличивающих путей при отсутствии двудольности: цветки. Утверждения о сжатии цветков. Алгоритм Эдмондса.

15. Модификации алгоритма Дейкстры для быстрого практического решения задачи о кратчайшем пути: «двухсторонний» алгоритм Дейкстры, использование landmarks (в случае неравенства треугольника).

16. Алгоритм Флойда—Уоршелла поиска кратчайших путей и циклов отрицательного веса.

17. Задача о потоке минимальной стоимости (и заданной величины). Два алгоритма: постепенная минимизация стоимости потока при неизменной величине; приращение величины за счёт минимально возможного приращения стоимости.

18. «Биологические» метаэвристики: генетические алгоритмы, алгоритмы муравьиных колоний. Иллюстрация на задачах shortest path и TSP.

19. Метод ветвей и границ.

20. Задачи исчерпывающего перебора сложных дискретных объектов. Подход Рида: упорядоченное перечисление. Метод обращения локального поиска Ависа—Фукуды.

21. Оптимизация во времени: онлайн-оптимизация. Задача «покупать/арендовать». «Задача секретаря». Анализ алгоритма кеширования LRU (Least Recently Used).

22. О понятии реоптимизации. Реоптимизация в метрической задаче коммивояжёра при увеличении веса одного ребра: NP-трудность точного решения, приближённый алгоритм с показателем аппроксимации 7/5 (что лучше, чем алгоритм Кристофидеса).

Билет 1:

1. Локальный поиск как широкий общий подход к решению задач дискретной оптимизации. Системы окрестностей. Пример системы окрестностей в задаче TSP: компромисс между силой окрестности и размером. Пример, в котором 2-окрестность не позволяет достичь глобального оптимума. Эвристика Кернигана—Лина: локальный поиск переменной глубины. Надстройки над локальным поиском: имитация отжига и табу-поиск.

2. Метод ветвей и границ.

Билет 2:

1. Дискретная линейная задача о подмножестве (DLS problem). Задачи TSP и MST как частные случаи DLS-задач минимизации; переход к максимизации. Наследственные системы. Базы и циклы. Ранг и нижний ранг множества, ранговый разброс. Матроиды: эквивалентные определения, примеры. Оценка качества работы жадного алгоритма на наследственной системе через её ранговый разброс. Следствие о корректности жадного алгоритма построения кратчайшего остовного дерева. Оценка рангового разброса через ограничение на число циклов. Субмодулярность ранговой функции матроида. Перечесение матроидов. Оценка числа циклов для наследственной системы через число матроидов в пересечении. Вероятность единственности решения задачи DLS при случайном выборе весов: лемма об изолировании.

2. Постановки задачи TSP в терминах ЦЛП. Условия Миллера—Таккера—Землина (полиномиальное количество неравенств в задаче TSP). Замечание «о некатастрофичности экспоненциального числа ограничений в задачах ЛП».

Критерии оценивания

- оценка «отлично (10)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;
- оценка «отлично (9)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;
- оценка «отлично (8)» выставляется студенту, показавшему всесторонние систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, и правильное обоснование принятых решений;
- оценка «хорошо (7)» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «хорошо (6)» выставляется студенту, если он знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «хорошо (5)» выставляется студенту, если он знает материал, и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;
- оценка «удовлетворительно (4)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «удовлетворительно (3)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет фрагментарно основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;
- оценка «неудовлетворительно (2)» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач;
- оценка «неудовлетворительно (1)» выставляется студенту, который не знает формулировок основных понятий дисциплины.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины.