МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

СОГЛАСОВАНО
Руководитель ОП
_____ Величко А.С.
(подпись) (ФИО)

УТВЕРЖДАЮ
И.о. директора департамента
Заболоцкий В.С.
(подпись) (ФИО)
«_28_» декабря 2021 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Аналитические сети
Направление подготовки 01.04.04 Прикладная математика
(Аналитические, социальные и экономические сети)
Форма подготовки: очная

курс 1 семестр 2
лекции 36 час.
практические занятия 0 час.
лабораторные работы 36 час.
в том числе с использованием МАО лек. 0 час. / пр. 0 час. / лаб. 36 час.
всего часов аудиторной нагрузки 72 час.
в том числе с использованием МАО 36 час.
самостоятельная работа 144 час.
в том числе на подготовку к экзамену 36 час.
контрольные работы (количество) 3
курсовой проект не предусмотрен
зачет не предусмотрен
экзамен 2 семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта по направлению подготовки 01.04.04 Прикладная математика, утвержденного приказом Минобрнауки России от 10.01.2018 г. № 15.

Рабочая программа обсуждена на заседании департамента математики, протокол № 6 от 28 декабря 2021 г.

И.о. директора департамента математики Заболоцкий В.С.

Составитель: профессор, канд. техн. наук, доцент А.Л. Абрамов

Владивосток
2021

Оборотная сторона титульного листа РПД

I. Рабочая программа пересмотрена на заседании департамента:

Протокол от «_____» _____ 20___ г. № _____

Директор департамента _____ _____
(подпись) (И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании департамента:

Протокол от «_____» _____ 20___ г. № _____

Директор департамента _____ _____
(подпись) (И.О. Фамилия)

III. Рабочая программа пересмотрена на заседании департамента:

Протокол от «_____» _____ 20___ г. № _____

Директор департамента _____ _____
(подпись) (И.О. Фамилия)

IV. Рабочая программа пересмотрена на заседании департамента:

Протокол от «_____» _____ 20___ г. № _____

Директор департамента _____ _____
(подпись) (И.О. Фамилия)

АННОТАЦИЯ

Дисциплина «Аналитические сети» предназначена для студентов направления подготовки 01.04.04 «Прикладная математика», магистерская программа «Аналитические, социальные и экономические сети».

Общая трудоемкость освоения дисциплины составляет 6 зачетных единиц (216 часов). Дисциплина реализуется на 1 курсе в 2-м семестре. Дисциплина входит в часть, формируемую участниками образовательных отношений, блока «Дисциплины (модули)». Дисциплина основана на знаниях, полученных студентом в курсах дискретной математики и теории графов и сетей.

Особенности построения курса: лекции (36 часов), лабораторные работы (36 часов), самостоятельная работа (108 часов), подготовка к экзамену (36 часов).

Содержание дисциплины охватывает следующий круг вопросов: экстремальные задачи для графов и сетей.

Цель – дать представление о моделях и подходах, применяемых при решении задач в практике бизнеса и экономики, основанных на теории графов и сетей.

Задачи:

- развитие способности знать специальные модели и методы решения задач в теории графов;
- развитие готовности использовать теоретические результаты по тематике дисциплины для анализа конкретных примеров из экономики и бизнеса;
- развитие готовности владеть стандартными инструментальными средствами решения типовых экстремальных задач на сетях и графах.

Для успешного изучения дисциплины «Аналитические сети» у обучающихся желательны следующие предварительные компетенции:

- готовность к самостоятельной работе.

В результате изучения данной дисциплины у обучающихся формируются следующие универсальные, общепрофессиональные, профессиональные

компетенции (элементы компетенций).

Код и наименование профессиональной компетенции	**Код ПС (при наличии ПС) или ссылка на иные основания**	**Код трудовой функции (при наличии ПС)**	**Индикаторы достижения компетенции**
Тип задач профессиональной деятельности: **научно-исследовательский**			
ПК-5 Способен к разработке и исследованию математических методов и моделей для проведения многовариантных аналитических расчетов и подготовки принятия решений	Анализ требований, предъявляемых к выпускникам	-	ПК-5.1. Формулирует модели, применяет методы анализа объектов, систем, процессов и технологий на основе математических моделей и методов прикладной математики ПК 5.2 Проводит сценарные аналитические расчеты для обоснования принимаемых решений по вариантам в том числе на основе программных средств

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
ПК-5.1. Формулирует модели, применяет методы анализа объектов, систем, процессов и технологий на основе математических моделей и методов прикладной математики	Знает равновесные и экстремальные задачи на сетях и графах в экономических, финансовых, социальных и информационных сетях, методы обоснования адекватности используемых моделей
	Умеет обнаруживать явления, моделируемые экстремальными постановками задач на сетях и графах в экономических, финансовых, социальных и информационных сетях, обосновывать адекватность используемых моделей
	Владеет методами решения равновесных и экстремальных задачи на сетях и графах в экономических, финансовых, социальных и информационных сетях, методами обоснования адекватности используемых моделей
ПК 5.2 Проводит сценарные аналитические расчеты для обоснования принимаемых решений по вариантам в том числе на основе программных средств	Знает алгоритмы решения равновесных и экстремальных задач на сетях и графах, методы оценки работоспособности и эффективности алгоритмов
	Умеет разрабатывать и реализовывать алгоритмы решения равновесных и экстремальных задач на сетях и графах» в экономических, финансовых, социальных и информационных сетях с помощью современных программных систем, оценивать работоспособность и эффективность алгоритмов
	Владеет методами проектирования и разработки алгоритмов решения равновесных и экстремальных задач на сетях и графах методами оценки работоспособности и эффективности алгоритмов

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Раздел I. Алгоритмы оптимизации на сетях и графах

Тема 1. Кратчайшие пути

Поиск кратчайшего пути между заданными вершинами в графе. Алгоритм Дейкстры поиска кратчайшего пути. Пример. Обоснование алгоритма Дейкстры. Экономические аспекты применения задачи. Поиск k кратчайших путей в графе. Алгоритм двойного поиска. Пример. Экономические аспекты применения задачи.

Тема 2. Остовные деревья в графе

Поиск всех остовных деревьев в графе. Алгоритм построения всех остовных деревьев в графе. Пример. Обонование алгоритма построения всех остовных деревьев в графе. Экономические аспекты применения задачи. Наикратчайшее остовное дерево. Алгоритм Краскала. Пример. Алгоритм Прима. Пример. Экономические аспекты применения задачи.

Тема 3. Гамильтоновы циклы

Поиск всех гамильтоновых циклов. Алгебраический метод. Алгоритм алгебраического метода. Пример. Экономические аспекты применения задачи. Наикратчайший гамильтонов цикл. Метод ветвей и границ. Алгоритм метода ветвей и границ. Пример. Экономические аспекты применения задачи.

Тема 4. Паросочетания и покрытия

Основные понятия и определения по теме “паросочетания и покрытия”. Паросочетание максимальной мощности. Алгоритм построения чередующегося дерева. Пример. Алгоритм построения паросочетания максимальной мощности. Пример. Обоснование алгоритма построения паросочетания максимальной мощности. Паросочетание максимального веса. Алгоритм Эдмондса – Джонсона. Пример. Построение от паросочетания к покрытию. Построение от покрытия к паросочетанию. Экономические аспекты применения задач.

Тема 5. Центры графов и сетей

Главный центр. Абсолютный центр. Главный абсолютный центр. Метод Хакими (нахождение абсолютного центра). Пример. Модифицированный метод Хакими (нахождение абсолютного центра). Пример. Итерационный метод (нахождение абсолютного центра). Пример. Экономические аспекты применения задачи. Главный абсолютный центр. Метод Хакими (нахождение

главного абсолютного центра). Пример. Модифицированный метод Хакими (нахождение главного абсолютного центра). Пример. Экономические аспекты применения задачи.

Тема 6. P-медианы графов и сетей

Поиск P-медианы является одним из наиболее широких классов задач, известного как минисуммные задачи расположения-распределения. Задача о нахождении p-медианы - это задача о размещении заданного числа пунктов обслуживания, при которых сумма кратчайших расстояний до ближайших пунктов принимает минимально возможное значение. Алгоритмы решения задачи поиска p–медианы (метод ветвей и границ, метод замены вершин, эвристические методы, генетические алгоритмы, поиск с запретами, метод имитации отжига, метод множителей Лагранжа). Алгоритм Тэйтца и Барта. Пример. Экономические аспекты применения задачи.

Тема 7. Потоки в сетях

Основные понятия и определения. Максимальный поток. Алгоритм поиска увеличивающей цепи. Пример. Алгоритм поиска максимального потока. Пример. Экономические аспекты применения задачи. Поток минимальной стоимости. Алгоритм поиска потока минимальной стоимости. Пример. Обоснование алгоритма поиска потока минимальной стоимости. Экономические аспекты применения задачи. Максимально динамический поток. Алгоритм поиска максимально динамического потока. Обоснование алгоритма поиска максимально динамического потока. Экономические аспекты применения задачи. Поток наискорейшего прибытия. Алгоритм поиска потока наискорейшего прибытия. Пример. Обоснование алгоритма поиска потока наискорейшего прибытия. Экономические аспекты применения задачи.

Раздел II. Задачи оптимизации и двойственность на графах

Тема 1. Основные теоретические сведения

Введение. Задачи оптимизации. Окрестности. Локальные и глобальные оптимумы. Элементы линейной алгебры. Основные понятия теории графов. Двойственная задача линейного программирования. Дополняющая

нежесткость. Лемма Фаркаша. Задача о кратчайшем пути и двойственная ей задача. Двойственный симплекс-алгоритм.

Тема 2. Прямо-двойственный алгоритм

Прямо-двойственный алгоритм. Прямо-двойственный метод в применении к задаче о кратчайшем пути. Прямо-двойственный метод в применении к задаче о максимальном потоке. Теорема о максимальном потоке и минимальном разрезе. Алгоритм пометок Форда-Фалкерсона. Алгоритм Флойда - Уоршелла. Прямо-двойственный метод в применении к задаче о потоке минимальной стоимости.

Тема 3. Применение к задачам о паросочетании

Алгоритм построения паросочетания в двудольном графе. Паросочение в двудольном графе и поток в сети. Паросочетание в произвольном графе - цветки. Паросочетание в произвольном графе – алгоритм. Взвешенное паросочетание. Венгерский метод для задачи о назначениях. Задача о взвешенном паросочетании в произвольном графе. Остовные деревья и матроиды. Задача и минимальном остовном дереве. Жадный алгоритм. Матроиды. Пересечение двух матроидов. Алгоритм отсекающей плоскости для задач целочисленного линейного программирования. Отсечение Гомори. Лексикографическое упорядочения. Конечность дробного двойственного алгоритма.

Тема 4. Приближенные алгоритмы, NP-полные задачи и трудно решаемые проблемы

Эвристики для задачи вершинном покрытии. Пример. Приближенные алгоритмы для задачи коммивояжера. Надежные сети минимальной стоимости. Топология прибрежной системы газопроводов. Равномерное разбиения графов. Деревья Штейнера. Временн*а*я сложность решения задач дискретной оптимизации. Основные классы сложности (P, NP, NPC). NP–трудные задачи (задача о рюкзаке, задача коммивояжера). Полиномиальные сведения. Теорема Кука. Другие NP-полные задачи: клика и задача коммивояжера, сочетание, покрытие, разбиение.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Лабораторные работы

Лабораторная работа №1. Поиск кратчайшего пути между заданными вершинами в графе

Лабораторная работа №2. Алгоритм Дейкстры поиска кратчайшего пути.

Лабораторная работа №3. Поиск k кратчайших путей в графе. Алгоритм двойного поиска.

Лабораторная работа №4. Поиск всех остовных деревьев в графе.

Лабораторная работа № 5. Наикратчайшее остовное дерево. Алгоритм Краскала. Алгоритм Прима.

Лабораторная работа №6. Поиск всех гамильтоновых циклов.

Лабораторная работа №7. Наикратчайший гамильтонов цикл.

Лабораторная работа №8. Паросочетание максимальной мощности.

Лабораторная работа №9. Паросочетание максимального веса.

Лабораторная работа №10. Абсолютный центр. Метод Хакими.

Лабораторная работа №11. Главный абсолютный центр. Модифицированный метод Хакими

Лабораторная работа №12. Алгоритмы решения задачи поиска p–медианы

Лабораторная работа №13. Максимальный поток.

Лабораторная работа №14. Поток минимальной стоимости.

Лабораторная работа №15. Задачи оптимизации, прямо-двойственный алгоритм и его применение.

Лабораторная работа №16. Применение к задачам о паросочетании.

Лабораторная работа №17. Остовные деревья и матроиды. Задача и минимальном остовном дереве.

Лабораторная работа №18. Приближенные алгоритмы, NP-полные задачи и трудно решаемые проблемы.

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Аналитические сети» представлено в Приложении 1 и включает в себя:

план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;

характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;

требования к представлению и оформлению результатов самостоятельной работы;

критерии оценки выполнения самостоятельной работы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

Контролируемые разделы дисциплины, этапы формирования компетенций, виды оценочных средств, зачетно-экзаменационные материалы, комплекты оценочных средств для текущей аттестации, описание показателей и критериев оценивания компетенций на различных этапах их формирования, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

1. Гданский, Н. И. Основы теории и алгоритмы на графах : учебное пособие / Н. И. Гданский. — Москва : ИНФРА-М, 2020. — 206 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-014386-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/978686.

2. Соколов, Г. А. Линейные целочисленные задачи оптимизации : учебное пособие / Г. А. Соколов. — Москва : ИНФРА-М, 2020. — 132 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-011144-5. - Текст : электронный. - URL: https://znanium.com/catalog/product/1106387.

3. Сдвижков, О. А. Практикум по методам оптимизации : учебное пособие / О. А. Сдвижков. - Москва : Вузовский учебник : ИНФРА-М, 2020. - 231 с. - ISBN 978-5-9558-0372-2. - Текст : электронный. - URL: https://znanium.com/catalog/product/1036460.

Дополнительная литература

1. Асанов, М. О. Дискретная математика: графы, матроиды, алгоритмы : учебное пособие / М. О. Асанов, В. А. Баранский, В. В. Расин. — 3-е изд., стер. — Санкт-Петербург : Лань, 2020. — 364 с. — ISBN 978-5-8114-4998-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/130477.

2. Костюкова, Н. И. Графы и их применение : учебное пособие / Н. И. Костюкова. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 147 с. — ISBN 978-5-4497-0367-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/89435.html.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Электронный ресурс «Единое окно доступа к образовательным

ресурсам». Форма доступа: http://window.edu.ru

Перечень дополнительных информационно-методических материалов

1. Виленкин Н. Я., Виленкин А. Н., Виленкин П. А. Комбинаторика. М: Фима: Изд-во Московского Центра непрерывного математического образования , 2010. – 400 с.

2. Харари Ф. Теория графов. М: URSS: Либроком, 2009. – 300 с.

3. Макаров И.М., Соколов В.Б., Абрамов А.Л. Целевые комплексные программы / Абрамов А.Л. — М.: Знание, 1980. – 136 с.

4. Dixit, A. and Stiglits, J. Monopolistic competition and optimum product diversity// American Economic Review 67, 1977. - 297-308 p.

5. Абрамов А.Л., Назарова И.Н. Моделирование экономического пространства и экономический анализ // Экономический анализ на дальнем востоке России, М.: Московский Общественный Научный Фонд, 2005. – 111 с.

6. Шевяхова Е.Ю., Рычков О.А. Новая экономическая география» и вопросы экономической политики. – М.: EERC, 2005. – 260 с.

7. Абрамов А.Л., Достовалов В.Н., Величко А.С., Давыдов Д.В. Моделирование экономического пространства и стратегическое развитие территорий // Стратегическое планирование на Дальнем Востоке: ответ на глобальные и локальные вызовы. - М.: Московский Общественный Научный Фонд, 2006. - 195с.

8. Krugman P. Increasing Returns and Economic Geography, Journal of Political Economics 99(3), 1991, p. 483-499

9. Fujita M., Krugman P. and Venables A.J. The Spatial Economy. London: MIT Press, 1999. – 680 p.

10. Кристофидес Н. Теория графов. Алгоритмический подход. - М.: Мир, 1978. – 360 с.

11. Майника Э. Алгоритмы оптимизации на сетях и графах. - М.: Мир, 1981. – 320 с.

12. Филипс Д., Гарсиа-Диас А. Методы анализа сетей. - М.: Мир, 1984. – 496 с.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Рекомендации по планированию и организации времени, отведенного на изучение дисциплины, описание последовательности действий обучающихся

Освоение дисциплины следует начинать с изучения рабочей учебной программы, которая содержит основные требования к знаниям, умениям и навыкам. Обязательно следует учитывать рекомендации преподавателя, данные в ходе установочных занятий. Затем – приступать к изучению отдельных разделов и тем в порядке, предусмотренном программой.

Получив представление об основном содержании раздела, темы, необходимо изучить материал с помощью рекомендуемой основной литературы. Целесообразно составить краткий конспект или схему, отображающую смысл и связи основных понятий данного раздела и включенных в него тем. Обязательно следует записывать возникшие вопросы, на которые не удалось ответить самостоятельно.

Подготовку к началу обучения включает несколько необходимых пунктов:

1) Необходимо создать для себя рациональный и эмоционально достаточный уровень мотивации к последовательному и планомерному изучению дисциплины.

2) Необходимо изучить список рекомендованной основной и дополнительной литературы и убедиться в её наличии у себя дома или в библиотеке в бумажном или электронном виде.

3) Необходимо иметь «под рукой» специальные и универсальные словари, справочники и энциклопедии, для того, чтобы постоянно уточнять

значения используемых терминов и понятий. Пользование словарями и справочниками необходимо сделать привычкой. Опыт показывает, что неудовлетворительное усвоение предмета зачастую коренится в неточном, смутном или неправильном понимании и употреблении понятийного аппарата учебной дисциплины.

4) Желательно в самом начале периода обучения возможно тщательнее спланировать время, отводимое на работу с источниками и литературой по дисциплине, представить этот план в наглядной форме (график работы с датами) и в дальнейшем его придерживаться, не допуская срывов графика индивидуальной работы и «аврала» в предсессионный период. Пренебрежение этим пунктом приводит к переутомлению и резкому снижению качества усвоения учебного материала.

Рекомендации по работе с литературой

1) Всю учебную литературу желательно изучать «под конспект». Чтение литературы, не сопровождаемое конспектированием, даже пусть самым кратким – бесполезная работа. Цель написания конспекта по дисциплине – сформировать навыки по поиску, отбору, анализу и формулированию учебного материала. Эти навыки обязательны для любого специалиста с высшим образованием независимо от выбранной специальности.

2) Написание конспекта должно быть творческим – нужно не переписывать текст из источников, но пытаться кратко излагать своими словами содержание ответа, при этом максимально структурируя конспект, используя символы и условные обозначения. Копирование и «заучивание» неосмысленного текста трудоемко и по большому счету не имеет большой познавательной и практической ценности.

3) При написании конспекта используется тетрадь, поля в которой обязательны. Страницы нумеруются, каждый новый вопрос начинается с нового листа, для каждого экзаменационного вопроса отводится 1-2 страницы

конспекта. На полях размещается вся вспомогательная информация – ссылки, вопросы, условные обозначения и т.д.

4) В итоге данной работы «идеальным» является полный конспект по программе дисциплины, с выделенными определениями, узловыми пунктами, примерами, неясными моментами, проставленными на полях вопросами.

5) При работе над конспектом обязательно выявляются и отмечаются трудные для самостоятельного изучения вопросы, с которыми уместно обратиться к преподавателю при посещении установочных лекций и консультаций, либо в индивидуальном порядке.

6) При чтении учебной и научной литературы всегда следить за точным и полным пониманием значения терминов и содержания понятий, используемых в тексте. Всегда следует уточнять значения по словарям или энциклопедиям, при необходимости записывать.

7) При написании учебного конспекта обязательно указывать все прорабатываемые источники, автор, название, дата и место издания, с указанием использованных страниц.

Подготовка к промежуточной аттестации по дисциплине: экзамену

К аттестации допускаются студенты, которые систематически в течение всего семестра посещали и работали на занятиях и показали уверенные знания в ходе выполнении практических заданий и лабораторных работ.

Непосредственная подготовка к аттестации осуществляется по вопросам, представленным в рабочей учебной программе. Тщательно изучите формулировку каждого вопроса, вникните в его суть, составьте план ответа. Обычно план включает в себя:

— определение сущности рассматриваемого вопроса, основных положений, утверждений, определение необходимости их доказательства;

— запись обозначений, формул, необходимых для полного раскрытия

вопроса;

— графический материал (таблицы, рисунки, графики), необходимые для раскрытия сущности вопроса;

— роль и значение рассматриваемого материала для практической деятельности, примеры использования в практической деятельности.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для осуществления образовательного процесса по дисциплине желательна учебная аудитория для проведения занятий лекционного типа.

Приложение 1

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

по дисциплине «Аналитические сети»

Направление подготовки 01.04.04 Прикладная математика

магистерская программа «Аналитические, социальные и экономические сети»

Форма подготовки очная

Владивосток
2021

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	4 неделя	Повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины.	22 часа	Собеседование
2	6 неделя	Самостоятельный разбор заданий и задач, решаемых на практических занятиях.	12 часов	Проект
3	10 неделя	Повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам лекций	20 часов	Собеседование
4	12 неделя	Самостоятельный разбор заданий и задач, решаемых на практических занятиях	12 часов	Проект
5	16 неделя	Повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам лекций	22 часа	Собеседование
6	18 неделя	Самостоятельный разбор заданий и задач, решаемых на практических занятиях	20 часов	Проект

Характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению

Раздел 1. Алгоритмы оптимизации на сетях и графах

Кратчайшие пути. Поиск кратчайшего пути между двумя заданными вершинами в графе. Каждой дуге (x, y) исходного графа G поставим в соответствие число $a(x, y)$. Если в графе G отсутствует дуга (x, y), то $a(x, y) = 0$. Назовем число $a(x, y)$ длиной дуги (x, y). Определим длину пути как сумму длин отдельных дуг, составляющих этот путь (цепь).

Для любых двух вершин s и t графа G могут существовать несколько путей, соединяющих вершину s с вершиной t. В данном пункте будет рассмотрен алгоритм, который определяет такой путь, ведущий из вершины s в вершину t, который имеет минимально возможную длину. Этот путь называется кратчайшим путем между вершинами s и t.

Пусть дан связный неориентированный взвешенный граф G. Задача о поиске наикратчайшей цепи состоит в том, чтобы найти подграф G_{st}, включающий наикратчайшие цепи от заданной вершины V_s до V_t, при этом сумма весов ребер минимальна. Данная постановка задачи формально описывается следующим образом:

$$\varphi : G = (V, A) \to \left[G_{st} = (V_{st}, A_{st}) \,|\, V_s, V_t \subset V, \sum_{\substack{l \in V_{st} \\ l \in \{1,2,\ldots,m\}}} c_l^{st} \to \min \right] \qquad \textbf{(1)}$$

Пусть дан связный неориентированный взвешенный без петель граф G. Задача о поиске наикратчайшей цепи состоит в том, чтобы найти подграф G_s, являющийся деревом кратчайших цепей, т.е. включающего кратчайшие цепи из V_s во все вершины графа. Данная постановка задачи формально описывается следующим образом:

$$\varphi : G = (V, A) \to [G_s = (V, A_s) \,|\, G_s - \textit{дерево кратчайших цепей}] \qquad \textbf{(2)}$$

Для решения этих задач в случае, если веса дуг неотрицательны, и в графе отсутствуют циклы отрицательного веса, разработан простой эффективный алгоритм, который носит название алгоритм Дейкстры.

Поиск k кратчайших путей в графе. Во многих случаях требуется знать не только кратчайшие пути между двумя заданными вершинами, но и вторые, третьи и т.д. по длине пути между этими же вершинами. В данном разделе описывается *алгоритм двойного поиска*, который находит k первых кратчайших путей из некоторой фиксированной вершины ко всем остальным вершинам исходного графа.

Остовные деревья в графе. Поиск всех остовных деревьев в графе. В некоторых ситуациях возникает необходимость в построении полного списка остовных деревьев графа G. Например, в том случае, когда надо отобрать "наилучшее" дерево, а критерий, позволяющий осуществить такой отбор, является очень сложным, так что непосредственное решение задачи оптимизации (не использующей перечисление всех остовных деревьев) оказывается невыполнимым.

Пусть дан граф G. Нужно найти в нем все остовные деревья:

$\varphi : G = (V, A) \to \{G_p = (V, A_p) \,|\, A_p \subseteq A, G_p - \textit{дерево}, p = |\, \beta_0 \cdot \beta_0^T \,|\}$, где β_0 - матрица инциденций исходного графа G с одной удаленной строкой.

Наикратчайшее остовное дерево. Эта постановка задачи звучит так: найти в связном взвешенном графе G (без петель) наикратчайшее остовное дерево. Такая задача может быть решена с помощью алгоритма Краскала или алгоритма Прима.

Гамильтоновы циклы. Поиск всех гамильтоновых циклов. Алгебраический метод

Этот метод включает в себя построение всех простых цепей с помощью последовательного перемножения матриц. Введем некоторые определения.

Внутреннее произведение вершин цепи $V_1,V_2,...,V_{k-1},V_k$ определяется как выражение вида $V_2 * V_3 * ... * V_{k-1}$. *Модифицированная матрица смежности* $B=\{\beta_{ij}\}$ - это $(n\times n)$ матрица, в которой $\beta_{ij}=\begin{cases} V_j, если\ \exists\ дуга\ (V_i,V_j) \\ 0, в\ противном\ случае \end{cases}$.

Предположим, что существует матрица $P_l=\{p_l(i,j)\}$, где $p_l(i,j)$ - сумма внутренних произведений всех простых цепей длины $l(l\geq 1)$ между вершинами V_i и V_j для $V_i \neq V_j$. Пусть $p_l(i,j)=0, \forall i$.

Алгоритм алгебраического метода построения всех гамильтоновых циклов в графе включает основной шаг алгебраического метода состоит в процедуре умножения матриц $B * P_l = P_{l+1}' = \{p_{l+1}'(s,t)\}$, где

$$p_{l+1}'(s,t)=\sum_k \beta(s,k) * p_l(k,t) \qquad \textbf{(1)}$$

т.е. $p_{l+1}'(s,t)$ является суммой внутренних произведений всех цепей из вершины V_s в вершину V_t длины $l+1$.

Наикратчайший гамильтонов цикл. Метод ветвей и границ. Пусть дан полный связный взвешенный граф G, необходимо выделить в нем подграф, содержащий гамильтонов цикл, сумма весов ребер в котором минимальна.

$$\varphi: G=(V,A)\rightarrow \begin{bmatrix} G_\varphi=(V,A_\varphi)\,|\,(V_0,V_1,...,V_{n+1}): \forall V_i\in V\ появляется\ в (V_0,V_1,...,V_{n+1})\ один\ раз \wedge \\ \sum_j c(V_1,V_2)\rightarrow \min, где\ A_j - ребра, входящие\ в (V_0,V_1,...,V_{n+1}) \wedge V_0=V_{n+1} \end{bmatrix}$$

Один из методов решения задачи построения наикратчайшего гамильтонова цикла – метод ветвей и границ.

Паросочетания и покрытия. *Паросочетание* – множество ребер такое, что каждой вершине графа инцидентно не более, чем одно ребро этого множества. Существует множество паросочетаний для любого графа, в связи с этим формулируются задачи поиска паросочетания максимальной мощности и максимального веса для взвешенного графа.

Покрытие – некоторое множество ребер в графе такое, что каждая вершина графа инцидентна по крайней мере одному ребру этого множества. Существует множество покрытий для данного графа. Если граф взвешенный, то для него формулируется оптимизационная задача поиска покрытия минимального веса.

Потоки. Основные положения по теме “потоки”. Поток определяет способ пересылки некоторых объектов из одного пункта в другой, т.е. из одной вершины графа в другую по его дугам (перемещение по дуге осуществляется в заданном на ней направлении). Вершина, из которой начинается перемещение объектов, называется *источником* и обозначается через s. Вершина, в которой заканчивается перемещение объектов, называется *стоком* и обозначается через t. Объекты, которые перемещаются из источника в сток, называются *единицами потока* или просто *единицами*. Если количество единиц потока, которое может проходить по дуге (x,y), ограничено, то говорят, что дуга (x,y) имеет ограниченную *пропускную способность*. Максимальную величину пропускной способности будем обозначать через $c(x,y)$. (В дальнейшем величина $c(x,y)$ будет называться просто пропускной способностью.). Сеть – это граф, в котором каждой дуге приписана некоторая пропускная способность.

Пусть $a(x,y)$ обозначает стоимость прохождения единицы потока по дуге (x,y), а $f(x,y)$ - количество единиц потока, проходящих по дуге (x,y). υ - это количество единиц потока, которое нужно переместить из источника в сток.

Обозначим через $i(x,y)$ максимальную величину, на которую может быть увеличен поток в дуге (x,y), а через $r(x,y)$ максимальную величину, на которую может быть уменьшен поток в дуге (x,y). Очевидно, что $i(x,y)=c(x,y)-f(x,y)$ и $r(x,y)=f(x,y)$.

Предположим, что мы хотим переслать дополнительное количество единиц потока из источника s в сток t. Существует три основных способа это сделать:

1) Этот способ можно использовать, если найден путь P из вершины s в вершину t, целиком состоящий из увеличивающих дуг, и величина дополнительного потока из s в t по пути P составит самое большее $\min\limits_{(x,y)\in P}\{i(x,y)\}$.

2) Этот способ можно использовать, если найден путь P из вершины t в вершину s, целиком состоящий из уменьшающих дуг. При этом можно было бы уменьшить поток в каждой дуге (x,y), что привело бы уменьшению потока из вершины t в вершину s и, следовательно, к увеличению чистого потока из вершины t в вершину s. Максимальное уменьшение потока вдоль пути P определяется величиной $\min\limits_{(x,y)\in P}\{r(x,y)\}$.

3) Этот способ является комбинацией предыдущих двух. Для реализации данного способа нужно найти цепь, соединяющую вершины s и t, дуги которой удовлетворяют следующим условиям: а) все *прямые* дуги цепи, имеющие направление от s к t, принадлежат множеству I; б) все *обратные* дуги цепи, имеющие направление от t к s, принадлежат множеству R. Если эти условия выполняются, то вдоль рассматриваемой цепи можно переслать дополнительный поток из s в t. Это осуществляется путём увеличения потока в прямых дугах, являющимися увеличивающими, и уменьшения потока в обратных дугах, являющимися уменьшающими. Максимальная величина дополнительного потока, который можно переслать вдоль соответствующей цепи из s в t, определяется как минимум из следующих двух величин:

$\min\{i(x,y):(x,y)-прямая\ дуга\}$

$\min\{r(x,y):(x,y)-обратная\ дуга\}$.

Минимальная из этих двух величин называется *максимальным увеличением потока* по соответствующей цепи. Каждая цепь из s в t любого из трёх рассмотренных выше типов, по которой могут быть дополнительно посланы единицы потока, называется *увеличивающей цепью.*

Максимальный поток. Для любого потока из s в t количество единиц, выходящих из любой вершины x (при этом $x\neq s, x\neq t$), должно быть равно количеству единиц, входящих в эту вершину, т.е.

$$\sum_{y\in X} f(x,y)-\sum_{y\in X} f(y,x)=0,(x\neq s,x\neq t). \qquad \textbf{(1a)}$$

(X - это множество всех вершин рассматриваемого графа). Кроме того, количество единиц потока, проходящих по каждой дуге (x,y), не должно превышать пропускной способности этой дуги, т.е.

$$0\le f(x,y)\le c(x,y),(x,y)\in A. \qquad \textbf{(2a)}$$

(A - множество всех дуг рассматриваемого графа). Суммарное число единиц потока, выходящих из источника, должно быть равно суммарному числу единиц потока, входящих в сток. Если указанное число обозначить через υ, то это условие можно представить следующим образом:

$$\sum_{y\in X} f(x,y)-\sum_{y\in X} f(y,s)=\upsilon, \qquad \textbf{(3a)}$$

$$\sum_{y\in X} f(y,t)-\sum_{y\in X} f(t,y)=\upsilon. \qquad \textbf{(4a)}$$

Любой поток из s в t должен удовлетворять всем четырём условиям (1a)-(4a). И наоборот, если может быть найден набор величин $f(x,y)$ при $(x,y)\in A$, для которого

выполняются эти четыре условия, то такой набор представляет собой поток из источника s в сток t.

Задача о максимальном потоке состоит в поиске потока, удовлетворяющего условиям (1а)-(4а), для которого величина υ максимальна. Алгоритм нахождения максимального потока был разработан Фордом и Фалкерсоном, и идея этого алгоритма состоит в следующем: выбирается некоторый начальный поток из s в t и с помощью алгоритма поиска увеличивающей цепи выполняется поиск увеличивающей цепи. Если он оказывается успешным, то поток вдоль найденной цепи увеличивается до максимально возможного значения. Затем выполняется поиск новой увеличивающей цепи и т.д. Если на каком-то этапе этой процедуры увеличивающую поток цепь найти не удаётся, выполнение алгоритма заканчивается: текущий поток из s в t является максимальным.

Поток минимальной стоимости. Задача о потоке минимальной стоимости может быть представлена следующим образом:

$$\min\left\{\sum_{(x,y)} a(x,y)f(x,y)\right\} \qquad \textbf{(1)}$$

При ограничениях:

$$\sum_{y}\left[f(s,y)-f(y,s)\right]=\upsilon$$

(2)

$$\sum_{y}\left[f(x,y)-f(y,x)\right]=0 \quad (x\neq s, x\neq t)$$

(3)

$$\sum_{y}\left[f(t,y)-f(y,t)\right]=-\upsilon$$

(4)

$$0\le f(x,y)\le c(x,y);\forall x,y$$

(5)

Сумма, входящая в соотношение (1) представляет собой общую стоимость потока. Уравнение (2) показывает, что чистый суммарный поток из источника s должен быть равен υ. Уравнение (3) показывает, что чистый поток из любой вершины x, не совпадающей ни с источником s, ни со стоком t, должен быть равен 0. Уравнение (4) показывает, что чистый поток из стока t равен $(-\upsilon)$. Условие (5) описывает требование, согласно которому поток в каждой дуге должен иметь величину, находящуюся в интервале от 0 до значения пропускной способности данной дуги.

Соотношение (1) можно заменить следующим образом:

$$\max\left\{p-\sum_{(x,y)} a(x,y)f(x,y)\right\}, \qquad \textbf{(6)}$$

где p - достаточно большое число (например, p может быть любым числом, большим максимальной стоимости прохождения единицы потока из источника s в сток t).

Алгоритм поиска потока минимальной стоимости решает задачу, состоящую в организации пересылки с минимальными затратами заданного количества υ единиц потока из источника в сток в графе с заданными на дугах пропускными способностями и стоимостями прохождения одной единицы потока.

Обзор моделей теории размещения (центры и p-медианы). Ниже представлены основные классические задачи локального поиска.

1. В простейшей задаче размещения (ПЗР) требуется разместить некоторые объекты, оказывающие услуги клиентам (например, фирмы) на множестве возможных пунктов $I=\{1,...,m\}$.

Пусть $J=\{1,...,n\}$ - множество клиентов, $c_{ij}\ge 0$ - издержки удовлетворения спроса j-

го потребителя услугой, предоставляемой объектом i. Стоимость открытия фирмы $f_i \geq 0,\ \forall i \in I$. Для решения задачи необходимо найти такое подмножество $I^* \subseteq I$, чтобы снабжение потребителей осуществлялось из фирм, обеспечивающих минимальные издержки.

$$\min_{I^* \subseteq I} \left\{ \sum_{i \in I^*} f_i + \sum_{j \in J} \min_{i \in I^*} c_{ij} \right\}$$

Данная задача является NP-трудной. Если представить ее в виде задачи целочисленного линейного программирования, то возможно найти решение с помощью эвристических методов.

Постановка рассматриваемой задачи может усложняться, например, учетом предпочтения клиентов, учетом числа предприятий, необходимых для предоставления одной услуги (многостадийная задача размещения).

2. Квадратичная задача назначения была предложена Koopmans and Beckman в 1957 г. Она предназначена для расположения объектов, когда стоимость размещения зависит от расстояния и взаимосвязи с другими объектами. Например, пусть $J = \{1, ..., n\}$ – множество размещаемых объектов, $I = \{1, ..., m\}$ – множество возможных размещений объектов, расстояние между местами возможных размещений - d_{kl}, *где* $k, l \in I$. Объем блага, перемещаемого из объекта i в объект j - b_{ij}, *где* $i, j \in J$.

Необходимо найти такое распределение объектов по возможным местам размещений, чтобы суммарный объем перевозок блага был минимальным.

Математическая модель:

$$\min \sum_{i \in J} \sum_{j \in J} \sum_{l \in I} \sum_{k \in I} d_{kl} b_{ij} x_{ik} x_{jl}$$

При ограничениях:

$$\sum_{i \in J} x_{ik} = 1,\ \forall k \in I;$$

$$\sum_{j \in J} x_{ik} = 1,\ \forall i \in J;$$

$$x_{ik} = \begin{cases} 1, \textit{если объект } i \textit{ размещается в месте } k, \\ 0 \textit{ иначе.} \end{cases}$$

Основным методом решения задач данного типа является эвристический метод ветвей и границ.

3. Задачи покрытия предполагают, что существует некоторое критическое расстояние или время покрытия, в течение которого фиксированные объекты должны быть обслужены, тогда такие объекты будут считаться "покрытыми". Впервые данную модель предложили Church и Revelle в 1974 г. Модели покрытия делятся на два основных типа: задачи размещения на деревьях и задачи размещения на сети. Основными областями применения являются коммутационные цепи, поиск данных, обеспечение ритмичности сборочного процесса, служебный персонал воздушных линий, размещение оборонных сетей, товарных складов. Пусть $I = \{1, ..., m\}$ - множество всех возможных пунктов для размещения объектов, $J = \{1, ..., n\}$ - множество размещаемых объектов.

$$a_{ij} = \begin{cases} 1, \textit{ если в пункте } i \textit{ можно разместить в } j; \\ 0 \quad \textit{иначе}. \end{cases}$$

Переменные:

$$x_i = \begin{cases} 1, \textit{ если в пункте } i \textit{ размещен объект}; \\ 0, \textit{ иначе.} \end{cases}$$

Математическая модель:

$$\min \sum_{i \in I} x_i$$

при ограничениях:

$$\sum_{i \in I} a_{ij} x_i \geq 1, \ j \in J;$$

$$x_i \in \{0,1\} \ i \in I.$$

4. Проблема поиска p-медианы является одним из наиболее широких классов задач, известного как минисуммные задачи расположения-распределения. Задача о нахождении p-медианы - это задача о размещении заданного числа пунктов обслуживания, при которых сумма кратчайших расстояний до ближайших пунктов принимает минимально возможное значение. Пусть $I = \{1, ..., n\}$ - множество всех возможных пунктов для размещения объектов, $J = \{1, ..., n\}$ - множество размещаемых объектов, d_{ij} - расстояние между $i \in I$ и $j \in J$, P – число размещаемых объектов, h_i - вес $i \in I$.

$$x_{ij} = \begin{cases} 1, \textit{ если пункт i покрывается из точки размещения объекта } j, \\ 0 \quad \textit{иначе.} \end{cases}$$

Переменные:

$$y_i = \begin{cases} 1, \textit{ если в пункте i размещен объект}; \\ 0, \textit{ иначе.} \end{cases}$$

Математическая модель:

$$\min \sum_i \sum_j h_i \, d_{ji} x_{ij}$$

при ограничениях:

$$\sum_j x_{ij} = 1, \ \forall i,$$

$$\sum_j y_i = p,$$

$$x_{ij} \leq y_j \ \ \forall i, j,$$

$$x_{ij}, y_j \in \{0,1\} \ \ \forall i, j.$$

В начале 20 века, Альфред Вебер представил проблему размещения предприятия между тремя точками с добавлением веса для каждой из точек, чтобы смоделировать потребительский спрос. Поиск средней точки соответствует нахождению лучшего места для объекта, чтобы удовлетворить спрос в точках. Эту задачу обобщили до проблемы нахождения медианы (средней точки) для n ≥ 3 точек на плоскости, а также до проблемы Вебера—нахождение нескольких объектов—случай поиска p> 1 медиан среди точек на плоскости.

Вебер размещал медианы (объекты) в евклидовом пространстве. В начале 60-х годов прошлого века Хакими разработал аналогичные задачи для нахождения медиан в сети или на графе, его задача о нахождении абсолютной медианы похожа на взвешенную задачу Вебера. Хакими определил абсолютную медиану, как точку на графе, которая дает минимальную сумму взвешенных расстояний между этой точкой и вершинами графа. Хакими позволил этой точке находиться в любой точке графа, но доказал, что оптимальная абсолютная медиана всегда находится в вершине графа, таким образом обеспечил дискретное представление непрерывной задачи. Хакими обобщил абсолютную медиану до p-медиан в целях сведения к минимуму суммы взвешенных расстояний. Эти точки могут находиться в любом месте вдоль ребер графа. Хотя не все оптимальные пути решения этой проблемы находятся в вершинах, Хакими показал, что всегда есть набор p вершин, который сводит к

минимуму значение целевой функции. Таким образом, Хакими повторно доказал возможность дискретного представления непрерывной задачи, ограничив поиск вершин. Решение, состоящее из p вершин графа, он назвал p-медианой графа.

Хакими разработал понятие абсолютной медианы и p-медианы, чтобы найти оптимальное расположение распределительного центра (центров) в коммуникационной сети. С этой работы задача о p-медиане стала неотделима от теории расположения, став одной из наиболее распространенных моделей размещения объектов.

Для решения оптимизационной задачи о p-медиане наиболее эффективными являются эвристические методы, находящие приближенное решение задачи.

5. В моделях p-центров требуется найти такое наименьшее покрывающее расстояние, что каждая вершина графа была бы покрыта. Задачи поиска абсолютных центров графа отличаются от задачи поиска центров тем, что размещаемые объекты могут располагаться в вершинах и на рёбрах графа, а не только в вершинах.

Первое решение задачи поиска абсолютного центра предложил S. L. Hakimi в 1964 году. В своей статье он показал, что оптимальное расположение центров всегда располагается в вершине графа (узле сети), а наилучшая позиция абсолютного центра не всегда совпадает с вершиной. Также, он предложил алгоритм, позволяющий найти абсолютные центры графа.

В 1965 г. S. L. Hakimi обобщил концепцию медиан во взвешенном графе к мультимедианной. Затем он же показал, что оптимальное распределение P-центров в сети находится в P-медиане соответствующего взвешенного графа. Метод поиска абсолютных центров, предложенный S. L. Hakimi является графическим, так как определение наилучших точек-претендентов происходит за счет сравнения на графиках расстояний точка-вершина для различных вершин графа. Кроме того, метод Хакими предназначен для решения задачи с одним абсолютным центром графа и не может быть обобщён на случай абсолютных p-центров. Для поиска p-центров Singer S. предложил применение эвристических методов для поиска приближенного решения, а Minieka E. для нахождения p-центров применил решения задачи о наименьшем покрытии.

Nicos C. и Peter V. предложили неграфический метод нахождения p-центров: итерационный алгоритм решения задачи об абсолютных p-центрах графа. Они показали, что данный метод является быстро сходящимся, поскольку поиск абсолютных центров можно закончить сразу же, как только достигнута необходимая точность в расположении центров и данный метод можно видоизменить таким образом, чтобы можно было находить решения, близкие к оптимальному, и, следовательно, проводить анализ устойчивости решения.

Hakimi S. L., Schmeichel E. F., Pierce J. G. представили некоторые улучшения и обобщения существующих методов для расчета нахождения абсолютных центров сети и P-центров деревьев. Была улучшена сходимость алгоритма Хакими: $O(mn^2 \log n)$ для взвешенных случаев и $O(mn \log)$ для невзвешенных случаев, где m-число ребер графа, а n - число вершин.

Drezner представил в 1984 году два эвристических алгоритма для задачи поиска p-центра. Они могут использоваться при решении других задач размещении-назначения.

Dyer and Frieze предложили в 1985 году жадный алгоритм, в котором первый центр выбирался случайным образом. Hochbaum and Shmoys в том же году представил двойной приближенный эвристический подход для решения задачи поиска p-центра.

Hochbaum и Pathria предложили в 1998 году задачу размещения p-центров на основных k-сетях, соответствующих k-периодам. Каждому периоду соответствует своя сеть, что необходимо для расположения постоянных размещаемых объектов. Целью задачи является минимизация максимальных расстояний по всем k-периодам.

Anderson и другие представили в 1998 году приближенный алгоритм, основанный на методе со спросом в множестве точек (demand point aggregation method).

Panigrahy and Vishwanathan представили в том же году решение для асимметричных p-

центров, неоднократно применяя алгоритмы покрытия.

Khuller and Sussmann предложили в 2000 году приближенный алгоритм для поиска p-центра с ограниченной пропускной способностью.

Bespamyatnikh и другие представили алгоритм в 2002 году, находящий за полиномиальное время решение задачи поиска p-центра в графе пересечений дуг окружностей.

Pallottino и другие в этом же году опубликовали локальный эвристический подход для поиска p-ценра с ограниченной пропускной способностью вершин.

P´erez-Brito рассмотрел P-Facility λ-Centdian задачу. Суть задачи заключается в поиске таких p точек, благодаря которым минимизируется выпуклая комбинация целевой функции p-медиан и p-центров. Например, размещение местного отделения банка может потребовать минимизации среднего расстояния, которое необходимо преодолеть всем потенциальными клиентами при условии, что для любого клиента отделение банка расположено не слишком далеко.

Задача k-Centrum Multi-Facility обобщает и объединяет p-центры и p-медианы. Цель данной объединяющей модели – минимизация суммы k наибольших расстояний до пунктов обслуживания. Задачи p-центров и p-медиан соответствуют случаю, когда k=1 и n , соответственно.

Burkard и Dollani ввели модель p-центров с положительными или отрицательными весами. Это модель p-центров, в которой вершины могут иметь как положительный, так и отрицательный вес, который определяет - благоприятен объект или нет.

Caruso и другие в 2003 году представили четыре алгоритма, два их них точечные и два эвристические. Они находят решение задачи поиска p-центра, основываясь на решении последовательности задач о покрытии. Два эвристических алгоритма могут решать задачу с 900 вершинами за несколько секунд.

Задача анти - p-центров (от англ. Anti P-Center Problem) была впервые представлена Klein и Kincaid. Вместо минимизации максимального взвешенного расстояния между требуемой вершиной и ближайшим до неё объекта обслуживания, в задачи анти-p-центра требуется максимизировать минимальное взвешенное расстояние между требуемой вершиной и объектом обслуживания. Другими словами, ищется p «нежелательных» объектов. Все веса вершин в этой модели являются отрицательными.

Основные методы решения задач теории размещения. Ниже представлены одни из самых известных методов решения NP-трудных задач: метод ветвей и границ (Branch-and-bound, BBM); последовательная линеаризация (SLP) и SQP; метод множителей Лагранжа (Lagrangian relaxation); метод имитации отжига (Simulated annealing); генетический алгоритм (Genetic algorithm); другие методы: дискретные методы, метод отсекающей плоскости, различные эвристики; смешанные (гибридные) методы.

1. Метод ветвей и границ был впервые предложен в 1960 году Ленд и Дойг для решения задач целочисленного программирования. Общая идея метода может быть описана на примере поиска минимума функции $f(x)$ на множестве допустимых значений переменной x. Функция f и переменная x могут быть произвольной природы. Для метода ветвей и границ необходимы две процедуры: ветвление и нахождение оценок (границ). Процедура ветвления состоит в разбиении множества допустимых значений переменной x на подобласти (подмножества) меньших размеров. Процедуру можно рекурсивно применять к подобластям. Полученные подобласти образуют дерево, называемое деревом поиска или деревом ветвей и границ. Узлами этого дерева являются построенные подобласти (подмножества множества значений переменной x). Процедура нахождения оценок заключается в поиске верхних и нижних границ для решения задачи на подобласти допустимых значений переменной x.

В основе метода ветвей и границ лежит следующая идея: если нижняя граница значений функции на подобласти A дерева поиска больше, чем верхняя граница на какой-либо ранее просмотренной подобласти B, то A может быть исключена из дальнейшего

рассмотрения (правило отсева). Обычно, минимальную из полученных верхних оценок записывают в глобальную переменную m. Любой узел дерева поиска, нижняя граница которого больше значения m, может быть исключен из дальнейшего рассмотрения.

Если нижняя граница для узла дерева совпадает с верхней границей, то это значение является минимумом функции и достигается на соответствующей подобласти.

2. Последовательная линеаризация. Приближенная замена нелинейных соотношений на линейные, нелинейных моделей на линейные, т. е. линеаризация соотношений, моделей и т. д. весьма распространена. Такая линеаризация обычно проводится в двух случаях: либо если эксперимент показывает , что отклонение от линейности в рассматриваемых диапазонах изменения переменных невелико и несущественно, либо же, если эти диапазоны малы и мы заменяем приращения переменных на их дифференциалы, отбрасывая члены высшего порядка малости.

3. Метод множителей Лагранжа. В начале семидесятых годов прошлого столетия была выдвинута одна, как оказалось, очень продуктивная идея для решения NP-трудных задач. Многие из них позволяют разбить систему ограничений задачи на две группы так, что удаление одной из них превращает задачу в полиномиально разрешимую. Эти ограничения заносятся в целевую функцию с некоторыми коэффициентами, множителями Лагранжа. Релаксированная таким способом задача дает оценку снизу на оптимум исходной задачи, если решается задача на минимум, и оценку сверху для задач на максимум.

4. Метод имитации отжига - общий алгоритмический метод решения задачи глобальной оптимизации, особенно дискретной и комбинаторной оптимизации. Один из примеров методов Монте-Карло. Алгоритм основывается на имитации физического процесса, который происходит при кристаллизации вещества, в том числе при отжиге металлов. Предполагается, что атомы уже выстроились в кристаллическую решётку, но ещё допустимы переходы отдельных атомов из одной ячейки в другую. Переход атома из одной ячейки в другую происходит с некоторой вероятностью, причём вероятность уменьшается с понижением температуры. Устойчивая кристаллическая решётка соответствует минимуму энергии атомов, поэтому атом либо переходит в состояние с меньшим уровнем энергии, либо остаётся на месте.

Алгоритм «имитации отжига» относится к классу пороговых алгоритмов локального поиска. На каждом шаге этого алгоритма для текущего решения в его окрестности выбирается некоторое решение и, если разность по целевой функции между новым и текущим решением не превосходит заданного порога, то новое решение заменяет текущее. В противном случае выбирается новое соседнее решение.

5. Генетический алгоритм - это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, напоминающих биологическую эволюцию.

На первом этапе необходимо построить начальную популяцию из K решений. Определить рекорд x^*, как лучшее решение в популяции. Пока не выполнении критерий остановки, выполняют следующий шаг: вычислить L потомков - выбрать два родительских решения, применить алгоритм скрещивания, к полученному решению с некоторой вероятностью применить алгоритм мутации, добавить новое решение популяции и изъять из популяции L наихудших решений, пересчитать рекорд x^*.

6. Основной идеей метода отсекающей плоскости является решение непрерывных задач, постепенно добавляя ограничения, которые приводят к целочисленным решениям по соответствующим переменным. Также существует множество алгоритмов, основой которых является деление исходной задачи на две более простые задачи, поставленные в тех классах, для которых имеются алгоритмы. Одна из таких подзадач дает верхнюю оценку, вторая нижнюю. Проверяется совместность задач и условие, что нижняя оценка меньше верхней.

Раздел 2. Задачи оптимизации и двойственность на графах

Задачи оптимизации на графах и теория двойственности. Многие задачи как практического, так и теоретического характера касаются выбора «наилучшей» конфигурации или множества параметров для достижения некоторой цели. За последние несколько десятилетий сложилась иерархия таких задач вместе с соответствующим набором методов решения. На одном конце этой иерархии находится общая задача нелинейного программирования:

Найти такое x, чтобы минимизировать f(x) при условии gi(x)>0 для i=1,2,…,m; hj(x)=0 для j=1,2,…,p, где f(x), gi(x), hj(x) – произвольные функции параметра x. Методы решения таких задач почти всегда итеративны по своей природе, и их сходимость изучается при помощи анализа действительного переменного.

Когда функция f выпукла, функции gi вогнуты и hj линейны, получается задача, которая называется задачей выпуклого программирования, обладающая тем свойством, что из локальной оптимальности ее решения следует оптимальность глобальная. Кроме того, для нее имеются условия оптимальности – условия Куна – Таккера.

Сделав следующий большой шаг, а именно, предполагая, что f и функции gi, hj всегда линейны, приходим к задаче линейного программирования. Для этого класса задач характерно несколько существенных отличий. Прежде всего любая задача из этого класса сводится к выбору решения их конечного множества возможных решений. Таким образом, эту задачу можно назвать комбинаторной. Указанное конечное множество кандидатов в решения представляет собой множество вершин выпуклого многогранника, определяемого линейными ограничениями.

Применения известного симплекс-метода приводит к оптимальному решению задачи линейного программирования за конечное число шагов. Этот алгоритм основывается на идее улучшения стоимости путем перехода от вершины к вершине в многограннике. Несколько десятилетий совершенствования привели к формам симплекс-алгоритма, которые в целом считаются эффективными – для них решение задачи с сотнями переменных и тысячами ограничений не вызывает затруднений. Однако справедливо также то, что имеются специально придуманные задачи, при увеличении размерности которых число шагов симлекс-алгоритма растет экспоненциально.

Сравнительно недавно был предложен алгоритм эллипсоидов для линейного программирования (ЛП), который гарантирует нахождение оптимального решения за число шагов, растущее как полином от размера задачи.

Некоторые задачи ЛП, а именно задачи о потоках и паросочетаниях, решаются намного эффективней общих задач ЛП. С другой стороны, эти задачи тесно связаны с задачами, которые трудноразрешимы. Например, задача о кратчайшем пути из вершины в вершину в графе лежит в классе задач о потоках и паросочетаниях, и для ее решения имеется алгоритм Дейкстры с полиномиальной сложностью. В противоположность этому задача поиска наикратчайшего гамильтонова цикла (задача коммивояжера), в которой ищется кратчайший завкнуты путь, проходящий через каждую вершины ровно один раз, принадлежит классу *NP*-полных задач, которые все не разрешимы с помощью полиномиальных алгоритмов.

Задачи о потоках и паросочетаниях можно рассматривать и как частный случай задач целочисленного линейного программирования. Это такие задачи ЛП, в которых ищутся решения наилучшей стоимости при условии целочисленности его координат. Как ив случае задач ЛП, для решения таких задач имеется конечный алгоритм. Однако на этом сходство кончается: общая задача целочисленного линейного программирования *NP*-полна. В нашей программе «Комбинаторной оптимизации» мы сначала изучим фундаментальных и доступных фактов о задачах выпуклого программирования. Затем изучим ЛП, исследуя симлекс-метод, его геометрию и алгоритмические следствия двойственности. Сделаем особый акцент на теоретико-графовых интерпретациях алгоритмов, что приводит к задачам о

потоках и паросочетаниях. Отсюда необходимо перейти к рассмотрению вопросов сложности и алгоритма эллипсоидов, а также к изучению *NP*-полных задач, которые должны дать представления о трудных комбинаторных задачах оптимизации.

Задачи оптимизации на графах и прямо-двойственный алгоритм. Для решения подобных задач применяется универсальный алгоритм Эдмодсона и Джонсона.

В алгоритме построения паросочетания с максимальным весом, предложенным Эдмодсоном и Джонсоном, используется алгоритм построения чередующегося дерева. Алгоритм построения паросочетания с максимальным весом называется также называется алгоритмом Эдмондса - Джонсона.

Введем несколько определений. Взвешенной увеличивающейся чередующейся цепью называется чередующаяся цепь, в которой общий вес ребер, не входящих в паросочетание, превышает общий вес ребер, входящих в это паросочетание. Первая вершина в этой цепи является открытой, если первое ребро в этой цепи не принадлежит паросочетанию. Последняя вершина в этой цепи является открытой, если последнее ребро в этой цепи не принадлежит паросочетанию. На рис. цепи C_1, C_2 и C_3 являются чередующимися.

Цепь C_1 называется слабой увеличивающей чередующейся цепью, если в ней число ребер, принадлежащих паросочетанию, превышает число ребер, не принадлежащих ему.

Цепь C_2 называется нейтральной увеличивающей чередующейся цепью, если в ней число ребер, принадлежащих паросочетанию, равно числу ребер, не принадлежащих ему.

Цепь C_3 называется сильной увеличивающей чередующейся цепью, если в ней число ребер, принадлежащих паросочетанию, меньше числа ребер, не принадлежащих ему.

Теорема: Паросочетание M тогда и только тогда является паросочетанием с максимальным весом, когда для него не существует взвешенных увеличивающих цепей.

Доказательство:

1) Если для паросочетания M имеется взвешенная увеличивающая цепь C, то M не может быть паросочетанием с максимальным весом. Паросочетание M', полученное при замене ребер цепи C, входивших в M, не входившими в него ребрами этой цепи, имеет больший вес, чем паросочетание M.

1) Для того, чтобы доказать вторую часть теоремы, предположим, что M^* - произвольное паросочетание с весом большим, чем вес паросочетания M. Для этого рассмотрим множество всех ребер, входящих только в одно из паросочетаний M или M^*.

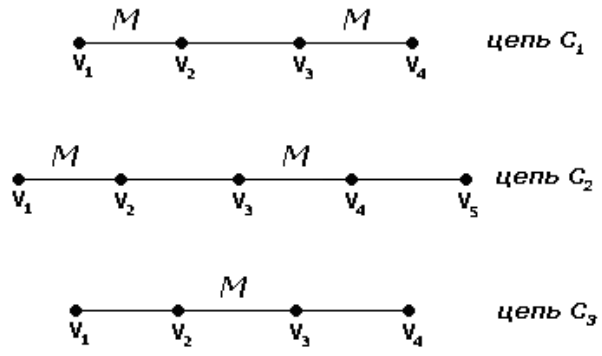

Рис. Взвешенные увеличивающиеся чередующиеся цепи

Каждый связный компонент, состоящий из ребер этого множества, должен быть либо циклом, либо цепью. Т.к. все циклы могут рассматриваться как цепи, то будем рассматривать каждый связный компонент, как некоторую цепь.

Т.к. паросочетание M^* имеет вес, превышающий вес паросочетания M, то в одной из этих цепей ребра, входящие в паросочетание M^*, должны иметь больший суммарный вес,

чем ребра, принадлежащие паросочетанию M . Следовательно, эта цепь должна быть взвешенной увеличивающей цепью для паросочетания M , что противоречит условиям теоремы. Теорема доказана.

Пусть V={V_1, V_2,…,V_z} – это множество всех подмножеств графа, включающих нечетное количество вершин. Пусть T_m - это множество ребер, обе вершины которых принадлежат подмножеству V_m и T={T_1, T_2, ….., T_z}. Обозначим количество элементов в множестве V_m как $2n_m+1$. Тогда имеет место следующая теорема.

Теорема: Ни одно паросочетание не может содержать более чем n_m ребер, принадлежащих множеству T_m.

Доказательство данной теоремы можно найти в [44].

Формальная постановка задачи ПС МВ. Пусть дан связный неориентированный взвешенный граф $G=(V,A)$, необходимо выделить в нем подграф, содержащий паросочетание максимального веса, т.е. подграф сумма весов ребер в котором максимальна:

$$\phi: G=(V,A) \rightarrow \begin{bmatrix} G_\phi = (V, A_\phi) | A_\phi \subseteq A \wedge \forall A_i, A_j \in A_\phi \wedge i \neq j, A_i \cap A_j = \emptyset \wedge \\ \sum_{A_j \in A} c(A_j) \rightarrow max \end{bmatrix}$$

Алгоритм Эдмондса - Джонсона

Шаг 1. (Выбор начальных значений). Пусть сначала ПС M_0 - это пустое множество ребер, а все двойственные переменные z_m равны нулю ($m = 1,2,\ldots,z$). Выбрать такие начальные значения для двойственных переменных y_i, $i \in X$, что $y_i + y_j \geq a(i,j)$ для всех ребер (i,j) (например, можно выбрать ребро, имеющее максимальное значение веса, и каждое y_i принять равным половине значения этого веса.). Положить $k := 0$ и обозначить исходный граф через $G_k = (X_k, E_k)$.

Шаг 2. (Поиск открытой вершины). Выбрать в графе G_k любую не фиктивную открытую вершину v со значением переменной $y_v > 0$. Если такой вершины не существует, то перейти к шагу 6. В противном случае, выделить в G_k множество E^* ребер (i,j), для которых выполняется условие:

$$y_i + y_j + \sum_{(i,j) \in T_m} z_m = a(i,j). \qquad (1)$$

С помощью алгоритма построения чередующегося дерева построить дерево с корнем в вершине v, содержащее только ребра из множества E^*. Если при этом найдется увеличивающая чередующаяся цепь, то перейти к шагу 3. Если будет построен нечетный цикл, то перейти к шагу 4. И, наконец, если будет построено венгерское дерево, то перейти к шагу 5.

Шаг 3. (Преобразование паросочетания). Этот шаг выполняется только тогда, когда с помощью алгоритма построения чередующегося дерева отыскивается увеличивающая цепь. Входящие в паросочетание M_k ребра цепи заменить не входящими в него ребрами этой же цепи. Вершина v перестает быть открытой. Вернуться к шагу 2.

Шаг 4. (Построение нечетного цикла). Этот шаг выполняется только при завершении алгоритма построения чередующегося дерева, когда найден нечетный цикл. Положить $k := k + 1$. Обозначить этот нечетный цикл через C_k. Стянуть C_k в фиктивную вершину a_k. Обозначить полученный новый граф через $G_k = (X_k, E_k)$. Пусть M_k — паросочетание, включающее все ребра, которые одновременно принадлежат паросочетанию M_{k-1} и графу

G_k. В дальнейшем при расстановке пометок все вершины, которые заменены фиктивной вершиной a_k, помечать одинаково с a_k. Вернуться к шагу 2 и продолжить построение чередующегося дерева с корнем в вершине, являющейся отображением вершины ν в G_k (даже если эта вершина является фиктивной). Отметим, что разметка вершин и раскраска ребер, полученных на последней итерации алгоритма построения чередующегося дерева, могут оказаться полезными на последующей итерации.

Шаг 5. (Венгерское дерево). Этот шаг выполняется только тогда, когда результатом работы алгоритма построения чередующегося дерева является венгерское дерево.

Принять

$$d_1 = \min\{y_i + y_j - a(i,j)\}, \qquad (2)$$

где минимум берется по всем (i, j), таким, что $i \in X_0$ является внешней вершиной дерева и вершина $j \in X_0$ и не помечена.

Положить

$$d_2 = \frac{1}{2}\min\{y_i + y_j - a(i,j)\}, \qquad (3)$$

где минимум берется по всем (i, j), таким, что $i \in X_0$ и $j \in X_0$ являются внешними вершинами дерева, но не стянуты в одну и ту же фиктивную вершину.

Принять

$$d_3 = \frac{1}{2}\min\{z_m\}, \qquad (4)$$

где минимум берется по всем множествам V_m вершин, мощность которых равна нечетному числу и которые стянуты в фиктивную вершину a_k, являющуюся внутренней вершиной дерева.

Положить

$$d_4 = \min\{y_i\}, \qquad (5)$$

где минимум берется по всем внешним вершинам $i \in X_0$ вершин дерева.

И, наконец, определить

$$d = \min\{d_1, d_2, d_3, d_4\}. \qquad (6)$$

Изменить двойственные переменные y_i, z_m следующим образом:

(а) Переменные y_i, соответствующие внешним вершинам дерева, уменьшить на величину d.

(б) Переменные y_i, соответствующие внутренним вершинам дерева, увеличить на величину d.

(в) Увеличить на $2d$ каждую двойственную переменную z_m, соответствующую внешней фиктивной вершине дерева в графе G_k.

(г) Уменьшить на $2d$ каждую двойственную переменную z_m, соответствующую внутренней фиктивной вершине дерева в графе G_k.

При этом если $d = d_1$, то в E^* включить ребро (i, j), для которого достигается минимум в (2). Это ребро теперь может быть окрашено в процессе реализации алгоритма построения чередующегося дерева. Поэтому следует вернуться к шагу 2 и продолжить построение чередующегося дерева с корнем в вершине ν.

Если $d = d_2$, то в E^* включить ребро (i,j), для которого достигается минимум в (3).

Это ребро теперь может быть окрашено в результате реализации алгоритма построения чередующегося дерева, что приводит к обнаружению нечетного цикла. Поэтому следует вернуться к шагу 2 и продолжить построение чередующегося дерева с корнем в вершине ν.

Если $d = d_3$, то какая-либо из двойственных переменных z_i становится равной нулю. Преобразовать в исходный нечетный цикл фиктивную вершину, соответствующую этой двойственной переменной. Положить $k := k + 1$. Полученный после обратного преобразования фиктивной вершины графа подграф обозначить через $G_k = (X_k, E_k)$. Положить паросочетание M_k, состоящим из всех ребер паросочетания M_{k-1} и n_i ребер множества T_i, инцидентных $2n_i$ вершинам множества V_i. Оставшаяся одна вершина множества V_i инцидентна ребру паросочетания M_k, так как все внутренние вершины дерева в графе G_{k-1} инцидентны ребрам паросочетания M_{k-1}. После такого преобразования чередующегося дерева следует вернуться к шагу 2 и продолжить построение чередующегося дерева с корнем в вершине ν.

Если $d = d_4$, то двойственная переменная y_i, соответствующая какой-либо из внешних вершин i, становится равной нулю. Тогда в чередующемся дереве цепь, соединяющая корень ν с вершиной i, является нейтральной увеличивающей цепью. В паросочетании M_k произвести замену входящих в него ребер цепи не входящими ребрами этой же цепи. Вершина ν становится инцидентной ребру паросочетания, а вершина i становится открытой, что допустимо из-за равенства нулю y_i. Вернуться к шагу 2.

Шаг 6. (Обратное преобразование фиктивных вершин в нечетные циклы). Этот шаг выполняется только после того, как на шаге 2 просмотрены все вершины, для которых не выполняется условие:

$$y_i > 0 \Rightarrow \sum_{j \in X} [x(i,j) + x(j,i)] = 1, \forall i \in X .$$

Рассмотреть все оставшиеся фиктивные вершины в полученном к этому моменту графе. Произвести преобразование фиктивных вершин в нечетные циклы в порядке, обратном их отысканию (полученная последней фиктивная вершина преобразовывается в нечетный цикл первой и т. д.) и образовать паросочетание максимальной мощности на каждом из этих нечетных циклов [32]. Полученное в результате паросочетание является паросочетанием с максимальным весом для начального графа G_0. На этом работа алгоритма заканчивается.

Алгоритм Эдмондса - Джонсона для задачи о минимальном покрытии. Формальная постановка задачи. Пусть имеет место граф G = (X,E) и пусть $V=\{V_1, V_2, \ldots, V_z\}$ – множество всех подмножеств множества X, состоящих из нечетного числа элементов. Пусть через $2n_m+1$ обозначено число вершин в подмножестве V_m, а через U_m – множество таких ребер, что хотя бы одна из инцидентных им вершин принадлежит X_m.

Справедливо следующее утверждение: каждое покрытие должно содержать в U_m по крайней мере (n_m+1) ребер.

Пусть a(i,j) – вес ребра (i,j). Пусть x(i,j)=1, если ребро (i,j) принадлежит покрытию; в противном случае x(i,j) = 0.

Рассмотрим следующую задачу линейного программирования:

$$\sum_{(i,j)} a(i,j)x(i,j) \to min \quad (1)$$

при ограничениях:

$$\sum_j [x(i,j) + x(j,i)] \geq 1 \text{ для всех } i \epsilon X \quad (2)$$

$$\sum_{(i,j) \epsilon U_m} x(i,j) \geq n_m + 1, m = 1,2, \ldots, z \quad (3)$$

$$x(i,j) \geq 0 \text{ для всех } (i,j) \quad (4)$$

В соответствии с ограничением (2) требуется, чтобы по крайней мере одно ребро, принадлежащее покрытию, было бы инцидентно вершине i. Ограничение (3) означает, что по крайней мере (n_m+1) ребер, принадлежащих множеству U_m, должно входить в покрытие.

Целевая функция (1) представляет собой общий вес покрытия. Любое покрытие

удовлетворяет ограничениям (2)-(4).

Задача, двойственная задаче (1)-(4), формируется следующим образом:

$\sum_{i\epsilon X} y_i + \sum_{m=1}^{z}(n_m + 1)z_m \rightarrow max$ (5)

при ограничениях

$y_i + y_j + \sum_{m:(i,j)\epsilon U_m} z_m \geq a(i,j)$ для всех (i,j) (6)

$y_i \geq 0$ для всех $i\epsilon X$ (7)

$z_m \geq 0, m = 1,2,\dots,z$ (8)

Условия дополняющей нежесткости для этой пары задач линейного программирования имеют следующий вид:

$x(i,j) > 0 \rightarrow y_i + y_j + \sum_{m:(i,j)\epsilon U_m} z_m = a(i,j)$ для всех (i,j), (9)

$$y_i > 0 \Rightarrow \sum_{j\in X}[x(i,j) + x(j,i)] = 1, \forall i \in X \quad (10)$$

т.е. вершина i покрывается только одним ребром.

$z_m > 0 \rightarrow \sum_{(i,j)\in U_m} x(i,j) = n_m + 1, m = 1,2,\dots,z.$ (11)

Таким образом, $2n_m+1$ вершин в X_m покрывается n_m такими ребрами, что обе инцидентные каждому из них вершины принадлежат X_m, и одним таким ребром, что одна инцидентная ему вершина принадлежит X_m.

Из ограничений (6)-(8) следует, что $y_i \leq \min\{a(i,j), a(j,i)\}$. Вершина i называется насыщенной, если y_i принимает наибольшее допустимое значение. В противном случае вершина i называется ненасыщенной. Если $y_i=0$, то вершина i называется пустой.

Отметим, что соотношение (10) не относится к пустым вершинам и, значит, что только пустые вершины могут быть инцидентны более чем одному ребру, принадлежащему покрытию. Каждая насыщенная вершина должна быть соединена ребром с пустой вершиной.

Алгоритм построения покрытие с минимальным весом

Этап 1. (Выбор паросочетания)

Шаг 1. (Выбор обозначений и начальных условий). Положить k:=0. Обозначить через $G_k = (X_k, E_k)$ рассматриваемый граф. Положить паросочетание M_k пустым (не содержащим ребер). Для сокращения записи формул вводится переменная

$w_i = y_i + \sum_{m:i\epsilon V_m} z_m$ для всех $i\epsilon X_0$ (12)

Принять двойственные переменные $z_m=0$, m=1,2,…,z.

Для двойственных переменных y_i выбрать любое значение, при котором выполнялись бы условия

$w_i + w_j \leq a(i,j)$ для всех $(i,j)\epsilon E_0$ (13)

$w_i \geq 0$ для всех $i\epsilon X_0$ (14)

Например, $y_i=0$ будет удовлетворять всем требованиям.

Шаг 2. (Просмотр открытой ненасыщенной вершины). Пусть $E^*=\{(i,j): w_i + w_j = a(i,j), i \in X_0, j \in X_0\}$. Выбрать любую, не являющуюся фиктивной, открытую ненасыщенную вершину $v \in X_k$, если таких вершин нет, то перейти к этапу 2.

В противном случае, используя алгоритм построения чередующегося дерева, на множестве ребер E^* для графа G_k построить чередующееся дерево с корнем в вершине v.

Если выполнение алгоритма построения чередующегося дерева заканчивается отысканием увеличивающейся цепи, то перейти к шагу 3.

Если работа этого алгоритма заканчивается отысканием нечетного цикла, то перейти к шагу 4.

Если алгоритм приводит к построению венгерского дерева, то перейти к шагу 5.

Шаг 3. (Построение увеличивающей цепи). Этот шаг выполняется только после того, как с помощью алгоритма построения чередующегося дерева обнаруживается увеличивающая цепь. В паросочетании M_k заменить ребра увеличивающей цепи на не принадлежащие ему ребрами этой же цепи. Открытая корневая вершина v становится инцидентной ребру паросочетания. Вернуться к шагу 2 для исследования другой открытой ненасыщенной вершины.

Шаг 4. (Построение нечетного цикла). Этот шаг выполняется только после того, как с помощью алгоритма построения чередующегося дерева найдется некоторый нечетный цикл. Положить k:=k+1. Стянуть найденный нечетный цикл в фиктивную вершину a_k. Граф, полученный в результате стягивания этого цикла в вершину, обозначить через G_k. Обозначить через M_k паросочетание, включающее все ребра ПС M_{k-1}, оставшиеся в графе G_k.

Вершины, входящие в стянутый нечетный цикл при всех последующих разметках по алгоритму построения чередующегося дерева, получают те же метки, что и фиктивная вершина a_k. Вернуться к шагу 2 и продолжить построения чередующегося дерева с корнем в вершине, являющейся отображением вершины v в графе G_k. В этом случае метки и окраска ребер последней итерации могут использоваться на следующей итерации алгоритма построения чередующегося дерева.

Шаг 5. (Венгерское дерево). Этот шаг выполняется только после того, как работа алгоритма построения чередующегося дерева заканчивается построением венгерского дерева.

Определить

$$d_1 = \min\{a(i,j) - w_i - w_j\}, \quad (15)$$

где минимум берется по всем внешним вершинам дерева $i \epsilon X_0$ и всем непомеченным вершинам $j \epsilon X_0$.

Положить

$$d_2 = \frac{1}{2} min\{a(i,j) - w_i - w_j\}, \quad (16)$$

где минимум берется по всем таким внешним вершинам дерева $i \in X_0$ и $j \in X_0$, что $(i,j) \notin E^*$

Определить

$$d_3 = \frac{1}{2}\min\{z_m\}, \quad (17)$$

где минимум берется по всем m, таким, что вершины подмножества V_m представляют внутреннюю фиктивную вершину дерева.

Положить

$$d_4 = \min\{a(i,j) - w_i\}, \quad (18)$$

где минимум берется по всем таким нефиктивным внешним вершинам дерева $i \in X_k$ и всем таким внутренним вершинам дерева $j \in X_0$, что $(i,j) \in E^*$

Определить

$$d_5 = \min\{w_i - \sum_{m:i \in V_m} z_m\}, \quad (19)$$

где минимум берется по вершинам $i \in X_0$, которые входят в нечетный цикл, стянутый во внешнюю фиктивную вершину дерева.

Определить

$$d = \min\{d_1, d_2, d_3, d_4, d_5\}. \quad (20)$$

Изменить двойственные переменные следующим образом:

1. Переменные w_i, соответствующие внешним вершинам дерева, увеличить на d.
2. Переменные w_i, соответствующие внутренним вершинам дерева, уменьшить на d.
3. Увеличить на 2d переменную z_m, соответствующую каждой внутренней фиктивной вершине дерева в графе G_k.
4. Уменьшить на 2d двойственную переменную z_m, соответствующую каждой внутренней фиктивной вершине дерева в графе G_k.

При этом если $d=d_1$, то к E^* добавляется ребро и процесс построения чередующегося дерева может быть продолжен. Вернуться к шагу 2 для продолжения построения чередующегося дерева с корнем в вершине v.

Если $d=d_2$, то к E* добавляется ребро, которое формирует нечетный цикл. Вернуться к шагу 2 для продолжения построения чередующегося дерева с корнем в вершине v.

Если $d=d_3$, то значение двойственной переменной для некоторой внутренней фиктивной вершины дерева снова становится равным нулю. Положить k:=k+1. Преобразовать обратно эту фиктивную вершину в исходный нечетный цикл. Она инцидентна одному из ребер паросочетания, так как имела метку внутренней вершины дерева. Включить в паросочетание

ребра нечетного цикла таким образом, чтобы все вершины этого цикла были инцидентны ребрам паросочетания. Принять полученный новый граф за Gk и новое паросочетание за Mk. Вернуться к шагу 2 для продолжения построения чередующегося дерева с корнем в вершине v.

Если $d=d_4$, то некоторая внешняя вершина j дерева становится насыщенной. Тогда чередующаяся цепь в чередующемся дереве, соединяющая корень v с вершиной j, является нейтральной увеличивающей цепью. Произвести замену ребер этой цепи, входивших в паросочетание, не входящими в него ребрами этой же цепи. Вершина j при этом становится открытой и насыщенной, а вершина v – инцидентной ребру, принадлежащему паросочетанию. Вернуться к шагу 2 и посмотреть другую открытую ненасыщенную вершину.

Если $d=d_5$, то y_i становится равной нулю для некоторой вершины i, содержащейся во внешней фиктивной вершине дерева a_m. Чередующаяся цепь от корня v до вершины a_m является нейтральной увеличивающей цепью. Произвести замену ребер цепи от v до a_m, входящих в паросочетание, не входящими в него ребрами этой же цепи. Вершина v становится инцидентной ребру, принадлежащему паросочетанию, а фиктивная вершина a_m – открытой. Вернуться к шагу 2 и просмотреть другую открытую ненасыщенную вершину.

Этап 2. (Преобразование паросочетания в покрытие).

Шаг 6. (Открытые нефиктивные вершины.) Этот шаг выполняется только после удаления на шаге 2 из графа G_k всех открытых ненасыщенных нефиктивных вершин. Для его выполнения необходимо включить в M_k ребра, соединяющие в G_k каждую открытую насыщенную нефиктивную вершину с пустой вершиной (пустая вершина не содержится ни в одной из фиктивных вершин, инцидентных ребру паросочетания). Теперь открытыми в G_k являются только фиктивные вершины. Перейти к шагу 7.

Шаг 7. (Обратное преобразование фиктивных вершин в нечетные циклы). Если в графе G_k отсутствуют фиктивные вершины, то M_k является покрытием с минимальным весом для графа G_0. В противном случае последнюю из оставшихся фиктивных вершин a_m необходимо преобразовать обратно в исходный нечетный цикл. Положить k:=k+1. Граф, полученный в результате преобразования фиктивной вершины, считать графом G_k.

Здесь возможны два случая:

а) в результате выполнения первого этапа алгоритма вершина a_m была покрыта ребром графа;

б) в результате выполнения первого этапа алгоритма вершина a_m оказалась открытой.

В случае (а) одна из вершин в цикле, соответствующем фиктивной вершине a_m, инцидентна ребру, принадлежащему паросочетанию, а остальные вершины в этом цикле являются открытыми. В M_k включаются все ребра, принадлежащие M_{k-1}, совместно с ребрами нечетного цикла, стянутого в a_m, которые инцидентны входящим в него открытым вершинам. Повторить шаг 7.

В случае (б) для некоторой вершины i нечетного цикла, соответствующего фиктивной вершине a_m, $y_i=0$. Положить M_k содержащим ребра, принадлежащие M_{k-1}, совместно с обоими ребрами нечетного цикла, инцидентными вершине i, и остальными ребрами этого нечетного цикла, принадлежащим паросочетанию. Повторить шаг 7.

Требования к представлению и оформлению результатов самостоятельной работы

Самостоятельная работа включает в себя повторение теоретического и практического материала дисциплины, заслушиваемого и конспектируемого в ходе аудиторных занятий; изучение основной и дополнительной литературы, указанной в рабочей учебной программе дисциплины, самоконтроль ответов на основные проблемные вопросы по темам занятий; самостоятельный разбор заданий и задач, решаемых на практических занятиях; самостоятельный повтор действий, осуществляемых в ходе выполнения лабораторных работ, в том числе при работе со специальным программным обеспечением.

Результаты самостоятельной работы представляются и оформляются в виде ответов на основные положения теоретического и практического материала дисциплины по темам; письменного разбора процесса решения практических заданий и задач; собственных действий, осуществляемых в ходе выполнения лабораторных работ.

В случае подготовки слайдов для защиты проекта, они должны быть контрастными (рекомендуется черный цвет шрифта на светлом фоне), кегль текста слайдов – не менее 22pt, заголовков – 32pt. Основная цель использования слайдов - служить вспомогательным инструментом к подготовленному выступлению, цитирование больших фрагментов текста на слайдах не допускается. Приветствуется использование рисунков, графиков, таблиц, интерактивного материала, однако, следует предусмотреть выбор цвета и толщину линий.

Слайды должны содержать титульный лист, цели и задачи (не более 2-х слайдов с обзором актуальности, новизны, теоретической и практической значимости работы), основные публикации с их кратким обзором (1-2 слайда), формальную постановку задачи и формулировку моделей (1-2 слайда), краткое тезисное (!) изложение ключевых положений работы (разумное количество слайдов с учетом общего времени выступления), заключение (с изложением результатов работы, подведением выводов, обсуждением практического использования работы, возможностей проведения дальнейших исследований и разработок в данной области).

Как правило, 12-15 слайдов оказывается достаточным для полного представления работы.

Критерии оценки выполнения самостоятельной работы

Общие критерии оценки выполнения самостоятельной работы – правильность ответов на вопросы по темам теоретической части дисциплины, верность получаемых ответов в ходе решения практических заданий и задач,

достижение правильного результата при осуществлении собственных действий по лабораторным работам.

Оценивание знаний в форме собеседования проводится по критериям:

- логичность изложения, знание и понимание основных аспектов и дискуссионных проблем по теме;
- владение методами и приемами анализа теоретических и/или практических аспектов по теме.

Оценивание знаний в форме проекта проводится по критериям:

- завершенность и полнота выполненных заданий в рамках проекта;
- владение методами и приемами решения конкретных задач и самостоятельность использования специализированного программного обеспечения;
- качество оформления письменного отчета в соответствии с правилами и стандартами оформления.