

Опыт проведения интерактивных занятий по курсу «Методы моделирования экономики»

А. М. Ревякин, И. В. Бардушкина, А. М. Терещенко

Национальный исследовательский университет «МИЭТ»

Отмечается особая важность и трудность для усвоения вопросов правильного выбора критерия при постановке задачи и оценке временной (комбинаторной) сложности алгоритма для ее решения. Предлагаются сценарии интерактивных занятий, в которых используется методика заранее запланированных неправильных решений, позволяющая формировать у обучающихся профессиональную компетентность, навыки научных логических рассуждений, умение ставить вопросы, анализировать решение проблем. На примерах решения задач оптимизации на графах и сетях анализируется жадный алгоритм. Приводятся примеры матроидов. Разбираются методы нахождения оптимальных решений в задачах размещения.

Ключевые слова: интерактивный метод обучения; комбинаторная сложность алгоритмов; NP -полнота; граф; сеть; дерево; паросочетание; оптимизация на графах и сетях; жадный алгоритм; матроид; остовы минимального и максимального веса; остов кратчайших расстояний; задача о назначениях.

Практический раздел курса «Методы моделирования экономики» для студентов экономического факультета содержит профессионально ориентированные задачи, в том числе алгоритмы решения оптимизационных задач на графах и сетях, где выбор правильной постановки и способа решения представляет самый сложный этап исследования. Каждая задача опирается на реальную экономическую проблему, следовательно, каждое занятие проходит в форме дискуссии [1]. В сценариях нескольких занятий используется методика заранее запланированных неправильных решений.

Тема занятия: полиномиальные оптимизационные алгоритмы на графах и сетях. Несколько студентов группы заранее подготавливают рефераты и в начале занятия делают обзор комбинаторной сложности алгоритмов и предлагают для

решения задач дискретной оптимизации использовать так называемый жадный алгоритм.

Полиномиальным алгоритмом (или алгоритмом полиномиальной временной сложности) называется алгоритм, у которого временная сложность равна Cn^k , где $k \geq 0$, а n — входная длина. Алгоритмы, временная сложность которых не поддается подобной оценке, называются *экспоненциальными*. Различие между этими двумя типами алгоритмов заметно при решении задач большого размера. Большинство экспоненциальных алгоритмов — это варианты *тривиального* алгоритма полного перебора. Принято считать, что задача *хорошо решается*, если для нее найден полиномиальный алгоритм.

Различия между полиномиальными и экспоненциальными алгоритмами могут быть иного характера, когда размеры

решаемых задач невелики. Переборные алгоритмы часто решают задачи малой размерности быстрее, чем полиномиальные. Более того, известны экспоненциальные алгоритмы, хорошо зарекомендовавшие себя на практике для решения индивидуальных задач [2; 3; 4]. Например: симплекс-метод для решения задач линейного программирования, метод динамического программирования для нахождения критического пути в задачах календарного сетевого планирования, метод ветвей и границ для задач о рюкзаке или о коммивояжере. Дело в том, что временная сложность определена как мера поведения алгоритма в наихудшем случае. Другими словами, алгоритм, имеющий временную сложность порядка 2^n , означает только то, что решение по крайней мере одной индивидуальной задачи размера n требует времени порядка 2^n . Однако большинство индивидуальных задач решается гораздо быстрее.

Рассмотрим задачу дискретной оптимизации. Пусть S — конечное множество, каждому элементу a которого приписан неотрицательный вес $w(a)$, $a \in S$. *Весом подмножества* $A \subseteq S$ называется сумма весов его элементов. Пусть I — некоторое семейство подмножеств множества S . Задача — выбрать из семейства I подмножество максимального веса.

Для решения подобных задач предлагается использовать так называемый **жадный алгоритм**. Алгоритм включает в себя четыре шага [2; 6; 7].

Шаг 1. Упорядочить множество S по убыванию весов так, чтобы

$$S = \{a_1, a_2, \dots, a_n\},$$

где $w(a_1) \geq w(a_2) \geq \dots \geq w(a_n)$.

Шаг 2. $A = \emptyset$, $i = 1$.

Шаг 3. Если $A \cup \{a_i\} \in I$, то $A = A \cup \{a_i\}$ и перейти к шагу 4.

Шаг 4. Если $i = n$ — конец алгоритма, иначе $i = i + 1$ — перейти к шагу 3.

Очевидно, что выходом жадного алгоритма является максимальное по включению множество из I .

С помощью жадного алгоритма студенты последовательно решают три задачи.

1. Найти остов максимального веса [2; 3; 4] в сети, заданной следующей матрицей смежности:

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>A</i>	—	6	10	10	—
<i>B</i>	6	—	5	—	—
<i>C</i>	10	5	—	10	7
<i>D</i>	10	—	10	—	4
<i>E</i>	—	—	7	3	—

2. Найти оптимальное назначение [2; 3; 4]. Пусть имеется три работника X_1, X_2, X_3 , претендующих на три должности Y_1, Y_2, Y_3 . Эффективность использования работника X_i на должности Y_j задана следующей матрицей:

	Y_1	Y_2	Y_3
X_1	5	1	2
X_2	1	4	1
X_3	2	2	3

Требуется распределить исполнителей таким образом, чтобы работы выполнялись наиболее эффективно. При этом соблюдаются условия: каждый исполнитель выполняет только одну работу; каждая работа может выполняться только одним исполнителем.

3. Найти *критический путь* (самый длинный путь) из вершины A в вершину D (см.: [2; 3; 4]) в сети G , заданной следующей матрицей смежности:

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>A</i>	—	7	4	—
<i>B</i>	—	—	3	9
<i>C</i>	—	—	—	10
<i>D</i>	—	—	—	—

Во всех задачах S — множество ребер сети, каждому ребру a приписан неотрицательный вес $w(a)$, $a \in S$. В первой задаче I — семейство всех подмножеств ребер, не содержащих циклы, во второй — множество паросочетаний графа, а в третьей — семейство всех дорог из вершины A в вершину D .

В результате применения жадного алгоритма получаем остов максимального веса 23, назначения X_1 на Y_1 , X_2 на Y_2 , X_3 на Y_3 — с эффективностью 12 и путь из A в D через вершины B, C длиной 20. Нетрудно убедиться, что все значения действительно являются решениями рассматриваемых задач.

Эффективность жадных алгоритмов вызвана тем, что элемент, один раз включенный в оптимальное решение, остается в нем до конца. Здесь нет проверок всех возможностей, характерных для алгоритмов с возвратом. Другими словами, жадный алгоритм является самым быстрым из всех возможных алгоритмов для решения конкретной задачи.

Далее студентам предлагается еще раз решить эти задачи с другими матрицами. Если решением задачи 1 с помощью жадного алгоритма для любой матрицы всегда является остов максимального веса, то с другими задачами сложнее. Например, если в задаче об оптимальном назначении использовать матрицу

	Y_1	Y_2	Y_3	
X_1	5	4	2	,
X_2	4	1	1	
X_3	2	1	2	

то жадный алгоритм даст такое же, что и для первой матрицы, назначение с эффективностью 8, а оптимальным будет назначение X_1 на Y_2 , X_2 на Y_1 и X_3 на Y_3 с эффективностью 10. Вывод: жадный алгоритм не всегда правильно решает задачу о назначениях.

Если в задаче «о критическом пути» изменить длину единственной дороги — из A в B — с 7 на 3, то жадный алгоритм найдет путь из A в D через вершину C длиной 14, хотя путь через вершины B, C очевидно длиннее (его длина 16). Заметим также, что если выбирать самый протяженный путь из каждой вершины, то его значение опять может оказаться меньшим.

В результате анализа решений рассмотренных задач возникает естественный вопрос, в каком случае можно гарантировать получение подмножества максимального веса при решении задачи с помощью жадного алгоритма. Ответ дает теорема Р. Радо и Ж. Эдмондса: если $M = (S, I)$ — матроид, то множество A , найденное с помощью жадного алгоритма, является независимым множеством M с наибольшим весом [6; 7].

Матроидом M на множестве S называется система $I \subseteq P(S)$ подмножеств конечного множества S , если выполняются следующие условия [7]:

- 1) $\emptyset \in I$;
- 2) если $A \subseteq B$ и $B \in I$, то $A \in I$;
- 3) если $A, B \in I$ и $|A| > |B|$, то найдется такое $a \in A \setminus B$, что $B \cup \{a\} \in I$.

Напротив, если (S, I) удовлетворяет условию наследственности (условие 2) и не является матроидом, то существует такая неотрицательная действительная функция $w(a)$, определенная на множестве S , что A не будет элементом семейства I с наибольшим весом.

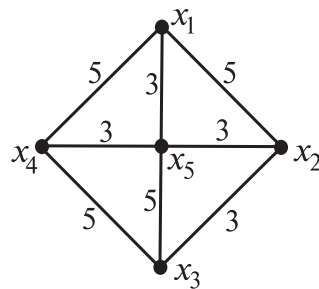
Эффективностью жадных алгоритмов обусловлено широкое применение матроидов в задачах дискретной оптимизации для получения быстрых алгоритмов.

Матроиды возникают в самых разнообразных комбинаторных и алгебраических контекстах. Понятия «независимость» и «базис» в векторных

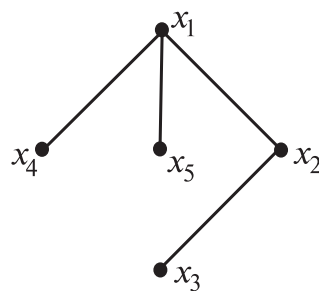
пространствах, «алгебраическая зависимость», «циклы» в графах, «поверхности» в проективных геометриях, «точечные полумодулярные решетки» сводятся к одной и той же матроидной структуре. Появление матроидов разнообразных форм во многих областях делает их объектами, достойными изучения.

Тема занятия: деревья, остовы, остовы кратчайших расстояний и минимального веса. Рассмотрим задачу проектирования сети дорог. Требуется населенные пункты соединить системой дорог так, чтобы любые два пункта были связаны либо непосредственно соединяющей их дорогой, либо через другие населенные пункты и чтобы общая длина была минимальной. В этой ситуации населенные пункты можно считать вершинами графа с весами ребер, равными длинам соединяющих эти населенные пункты дорог. Тогда искомая сеть будет остовым подграфом рассматриваемого графа. Очевидно, что этот подграф должен быть деревом. Поскольку полный граф K_n содержит n^{n-2} различных остовов, то решение этой задачи «слепым» перебором вариантов потребовало бы чрезвычайно больших вычислений даже при относительно малых n . Однако для ее решения имеются эффективные алгоритмы. Например, алгоритм Дейкстры — алгоритм нахождения кратчайших путей из одной вершины сети во все остальные, применимый к произвольному связному графу. Студентам предлагается самим задать сеть, построить в ней остов кратчайших расстояний из фиксированной вершины во все остальные. Если веса интерпретировать как стоимость дорог, возникает вопрос, является ли данный остов ответом на задачу о минимальной стоимости строительства дорог. На рисунке для заданного связного неориентированного графа G с указанными

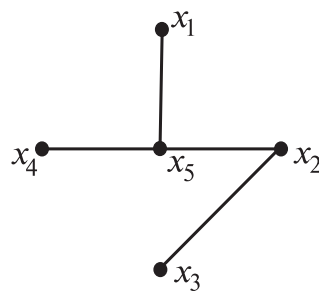
весами ребер (а) построены дерево кратчайших путей из вершины x_1 (б) и остов минимального веса (в).



а)



б)



в)

Связный граф G (а); дерево кратчайших путей из x_1 (б); остов минимального веса графа G (в)

Вопросы для организации дискуссии, побуждающие студентов к высказыванию идей. Каждый следующий вопрос формулируется после получения и обсуждения ответа на предыдущий вопрос.

1. Является ли найденный остов самым дешевым проектом?
2. Может ли построение остова кратчайших расстояний из другой вершины удешевить проект?

3. Как выбрать другую вершину с целью сделать проект более дешевым?

Предлагается построить остов минимального веса с помощью одного из трех алгоритмов, двойственных жадному алгоритму (алгоритмы Краскала, Борувки и Прима [3]), и сравнить стоимость проектов.

Предложим студентам самостоятельно убедиться в том, что алгоритмы Краскала, Борувки и Прима двойственны жадному алгоритму и тесно связаны с матроидами, задаваемыми графами и сетями.

В результате осмысления полученных результатов приходим к выводу: остов минимального веса является наиболее дешевым проектом. Обсуждаем важность правильного выбора критерия минимизации проекта. Формулируем задачи, для решения которых остов кратчайших расстояний предпочтительней, чем остов минимального веса.

Тема занятия: методы нахождения в сетях центров и медиан. Занятие начинается с докладов студентов о центрах и медианах в сетях. На практике постоянно возникают задачи *наилучшего* размещения оборудования (или средств обслуживания), решаемые на сетях или графах. В частности, если граф представляет сеть дорог, и вершины соответствуют отдельным районам, то можно поставить задачу оптимального размещения больниц, полицейских участков, пожарных частей и многих других, крайне необходимых, предприятий и служб. В таких случаях критерий оптимальности может состоять в минимизации расстояния (или времени проезда) от пункта обслуживания до самой отдаленной вершины графа, т. е. в оптимизации *наихудшего* варианта. В более общей задаче требуется разместить несколько таких пунктов обслуживания. При этом самая

отдаленная вершина графа должна находиться на минимально возможном расстоянии по крайней мере от одного пункта обслуживания. К таким задачам относится размещение аварийных служб, а объективным требованием является минимизация наибольшего расстояния от произвольной вершины графа до ближайшего к ней пункта обслуживания. Задачи данного типа называются *минимаксными задачами размещения*. Места размещения пунктов обслуживания, полученные при решении минимаксной задачи, называются *центрами* графа.

В некоторых задачах размещения предпочтительно было бы минимизировать сумму всех расстояний от вершин графа до центра обслуживания (если предполагать, что требуется найти место для размещения только одного пункта обслуживания). Рассматриваемый критерий является наиболее подходящим, например, для задачи размещения склада в сети дорог, где вершины представляют потребителей, обслуживаемых этим складом, или для задачи размещения телефонных станций в телефонной сети, где вершины представляют абонентов. Задачи данного типа относятся к *минисуммным задачам размещения*, однако целевая функция часто является не только суммой расстояний, но и суммой различных функций от расстояний. Места размещения пунктов обслуживания, полученные при решении минисуммной задачи, называются *медианами* графа.

Далее предлагается обсудить и выбрать критерии для оптимизации задач размещения на примере округа, состоящего из шести городов.

Задача 1. Администрация округа решила построить новый пост МЧС, обслуживающий все шесть городов. Пост должен располагаться возле одной

из автомагистралей, но таким образом, чтобы минимизировать расстояние до наиболее отдаленного от нее города. Если автомагистраль округа изображается ребром графа, то задача размещения пожарного поста становится задачей такого размещения точки на ребре графа, при котором минимизируется расстояние вдоль ребер (автомагистралей) от этой точки до наиболее удаленной вершины графа (до города).

Задача 2. Необходимо разместить здание новой почты в округе таким образом, чтобы сумма расстояний от почты до каждого города была минимальной.

Задачи 1 и 2 по существу одинаковые. Они различаются только критериями оценки качества размещения. В задаче 1 минимизируется максимальное расстояние, а в задаче 2 — сумма расстояний. Точка размещения, выбранная в соответствии с первым критерием, — точка (центр), в которой минимизируется максимальное расстояние до всех вершин графа. Точка размещения, выбранная в соответствии со вторым критерием, — медиана, в которой минимизируется сумма расстояний до всех вершин графа.

Задача 3. В том же округе разместим станцию эвакуаторов для оказания помощи водителям, попавшим в аварию. Критерием качества размещения станции будет минимум максимального расстояния, которое эвакуатор должен преодолеть до возможного места аварии. В этом случае вместо максимального расстояния до всех вершин графа рассматриваем максимальное суммарное расстояние до всех точек каждого ребра сети.

Задача 4. Место размещения телефонной станции должно находиться либо в одной из вершин сети, либо вдоль какой-либо автомагистрали. Место размещения следует выбрать таким образом, чтобы минимизировать суммарную

протяженность всех телефонных линий, которые будут проложены между станцией и шестью городами. Усложним задачу: пусть города имеют различную численность населения, и количество требуемых линий между станцией и каждым из городов находится в пределах от 1 до m . Заметим, что если для соединения каждого города с телефонной станцией требуется только одна линия, то сформулированная задача имеет математическую структуру, аналогичную структуре задачи 2. Однако в сложном варианте следует учитывать численность населения конкретного города и влияние на выбор места размещения станции более густонаселенных городов.

В заключение занятия на основе рассмотренных примеров формулируются выводы по выбору метода решения задач оптимизации.

Литература

1. Ревякин А. М., Бардушкина И. В., Бардушкин В. В. Сценарии проблемных лекций по линейной алгебре, проводимых в интерактивных формах // Вторые Всероссийские Декартовские чтения «Декартовский рационализм и современная наука»: мат.-лы науч.-практ. конф. (17 апр. 2015, Москва, Зеленоград). М.: МИЭТ, 2015. С. 159—169.
2. Ревякин А. М., Бардушкина И. В. Математические методы моделирования в экономике. М.: МИЭТ, 2013. 327 с.: ил.
3. Математические методы принятия решений и сетевые модели в управлении и экономике / И. Н. Абанина, В. В. Бардушкин, И. В. Бардушкина и др.; под общ. ред. И. Н. Абаниной, А. М. Ревякина. М.: МГАДА, 2014. 176 с.: ил.
4. Ревякин А. М. Сетевые модели для принятия решений // Вестник Московского городского педагогического университета. Серия: Экономика. 2013. № 3 (24). С. 173—182.
5. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982. 416 с.: ил.
6. Исаченко А. Н., Ревякин А. М. Матроиды в математическом моделировании экономических систем // Экономические и социально-гуманитарные исследования. 2015. № 1 (5). С. 13—18.

7. *Revyakin A. M.* Matroids // Journal of Mathematical Sciences. 2002. Vol. 108. № 1. P. 71—130.

Ревякин Александр Михайлович — кандидат физико-математических наук, доцент кафедры высшей математики № 2 (ВМ-2) МИЭТ. E-mail: arevyakin@mail.ru

Бардушкина Ирина Вячеславовна — кандидат физико-математических наук, доцент кафедры ВМ-2 МИЭТ. E-mail: i_y_bars@mail.ru

Терещенко Анатолий Михайлович — доктор технических наук, профессор кафедры ВМ-2 МИЭТ. E-mail: 4443655@mail.ru