Модуль 3. NP-трудные задачи

Лекция 13

Задача коммивояжёра.
Метод ветвей и границ.

План лекции

- Задача коммивояжёра
  - Гамильтоновы циклы
  - Задача коммивояжёра
  - Переборный алгоритм
- Метод ветвей и границ
  - Общий принцип
  - МВГ для задачи коммивояжёра
    - ✓ Варианты ветвления
    - ✓ Варианты оценки

Что делать?

1) Решать долго (за экспоненциальное время)
2) Пытаться сократить время решения для входов, ожидаемых на практике
   – Полиномиальные в среднем алгоритмы
   – Параметризованные алгоритмы
3) Решать приближённо
   – С гарантированной *оценкой точности*
   – Без гарантий, но как правило достаточно точно
   – С некоторой *вероятностью* ошибки

Задача коммивояжёра

Определения

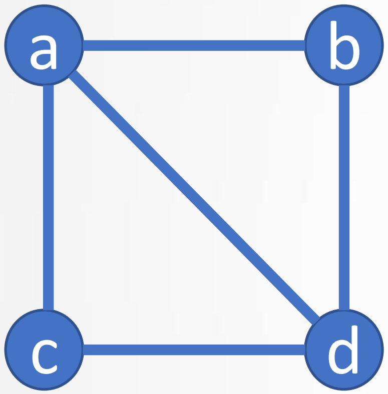
Рассмотрим $G(V,E)$ – связный граф.

Определения

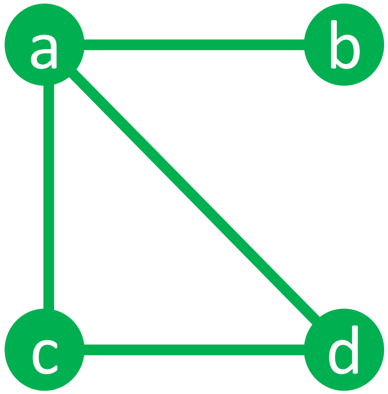
Цикл Z (путь P) называется ***гамильтоновым циклом (путём)*** на G, если и только если Z (P) проходит ровно один раз через каждую вершину графа G.

Граф $G(V,E)$ называется *гамильтоновым* (*полугамильтоновым*), если на нём существует гамильтонов цикл (путь).

Определения

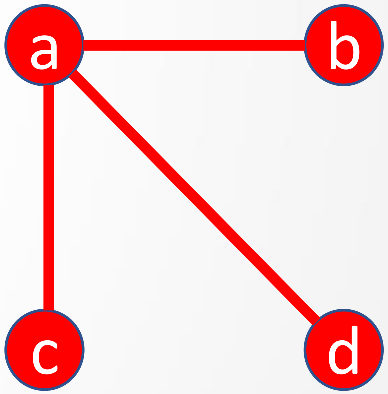


Гамильтонов граф



Полугамильтонов граф

Негамильтонов граф



Определения

Пусть $G(V,E)$ – связный граф; $w: E \longrightarrow R_+$ - весовая функция.

Вес цикла Z (пути P) определим как $w(Z) = \sum_{e \in Z} w(e)$.

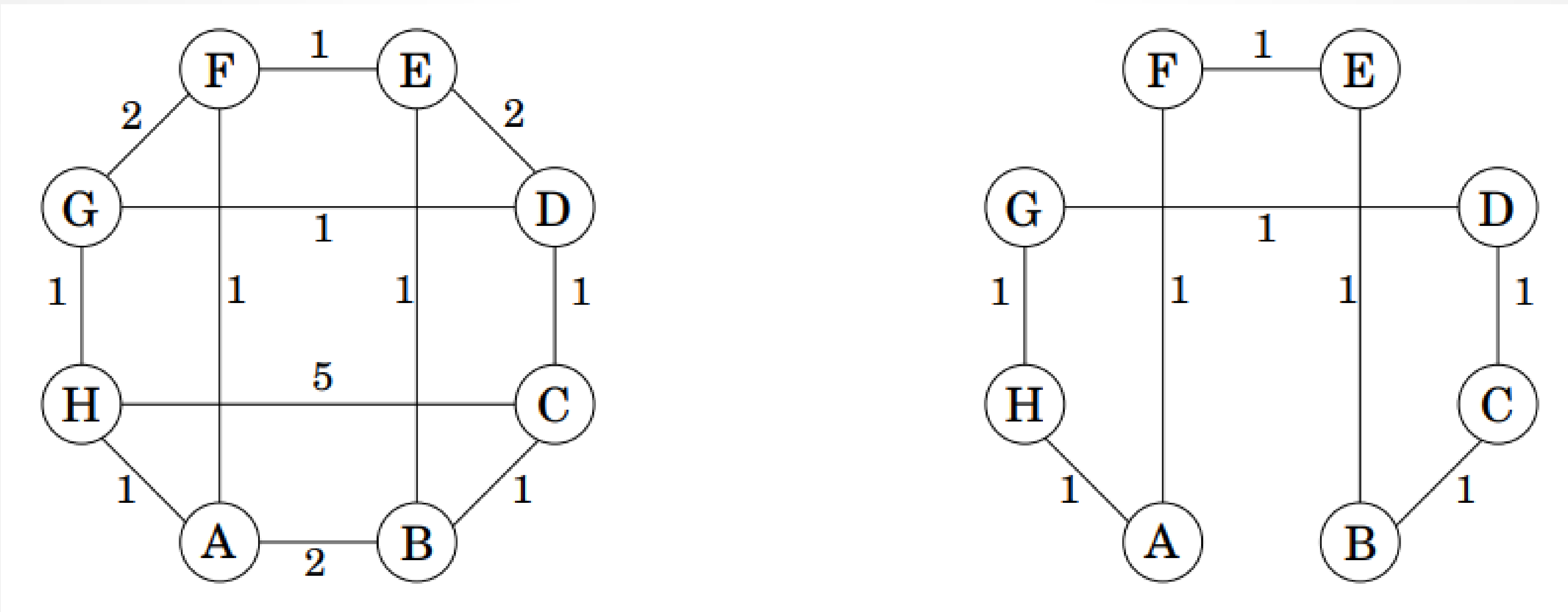
Распознавательная задача: является ли граф $G(V,E)$ (полу)гамильтоновым?

Вычислительная задача: построить на $G(V,E)$ гамильтонов цикл (вернуть ‘NULL’, если $G(V,E)$ не гамильтонов).

Оптимизационная задача (=задача коммивояжёра, Travelling Salesman Problem, TSP): построить минимальный гамильтонов цикл на графе $G(V,E)$ (вернуть ‘NULL’, если $G(V,E)$ не гамильтонов).

Определения

Пример графа и оптимального гамильтонова цикла на нём:



http://algorithmics.lsi.upc.edu/docs/Dasgupta-Papadimitriou-Vazirani.pdf

Решение ЗК

Теорема 1: Задача коммивояжёра является NP-трудной.

Что делать?

1) Решать долго (за экспоненциальное время)

   – Пытаться сократить перебор за счёт доп. анализа (*метод ветвей и границ*)

1) Пытаться сократить время решения для входов, ожидаемых на практике

   – Полиномиальные в среднем алгоритмы

   – Параметризованные алгоритмы

3) Решать приближённо

   – С гарантированной *оценкой точности*

   – Без гарантий, но как правило достаточно точно (*эвристики*)

   – С некоторой *вероятностью* ошибки

Решение ЗК

Определение: Задача коммивояжёра называется *метрической* (MTSP), если весовая функция $w: E \longrightarrow R_+$ является метрикой, то есть удовлетворяет аксиомам метрики:

1) $w(u,v) \geq 0$; $w(u,v)=0 \Leftrightarrow u=v$

2) $w(u,v) = w(v,u)$

3) неравенство треугольника: $w(u,v) \leq w(u,x) + w(x,v)$

Решение ЗК

Метрическая ЗК является важным частным случаем общей ЗК.

Важным частным случаем метрической ЗК является евклидова ЗК: вершины графа представляют точки в R^n, а вес ребра = евклидову расстоянию между вершинами.

Решение ЗК

Теорема 2: Метрическая задача коммивояжёра является NP-трудной.

Теорема 3: Даже евклидова задача коммивояжёра является NP-трудной.

Решение ЗК

Рассмотрим переборное (=точное, но медленное) решение общей ЗК.

Решение задачи (гамильтонов цикл или путь) можно представить минимум двумя способами:

1) Перестановка вершин: список вершин в том порядке, в котором через них проходит цикл/путь.
2) Последовательность рёбер: список рёбер в том порядке, в котором через них проходит цикл/путь.

Представление в виде перестановки вершин немного удобрее, т.к. для проверки корректности достаточно проверить, что соседние в перестановке вершины смежны на графе (включая первую и последнюю вершины в случае замкнутой задачи коммивояжёра). При использовании последовательности рёбер проверка корректности выполняется сложнее.

Решение ЗК

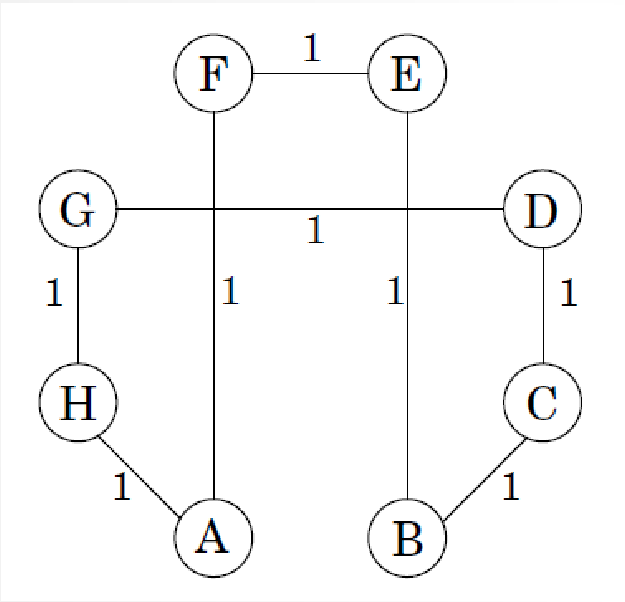
Алгоритм решения с помощью перебора ***перестановок***:

- Пусть $n = |V|$.
- Контур можно задать как перестановку номеров вершин.
- Но считаем, что в конце всегда стоит v_n.
- Поэтому перебираем только перестановки чисел $\{1, \ldots, n\text{-}1\}$.

Временная сложность: $O((n-1)!)$.

Решение ЗК

Для поиска оптимального *цикла* нужно сгенерировать $(n-1)!$ перестановок.



AFEBCDGH

FEBCDGHA

EBCDGHAF

…

Для неориентированного графа: только $\frac{(n-1)!}{2}$:

AHGDCBEF

HGDCBEFA

…

Генерация перестановок

Вызов: ProcessPermutations(*A*,*n*)

<u>ProcessPermutations(*A*,*k*)</u>

if *k* = 1 then Process(*A*)

else

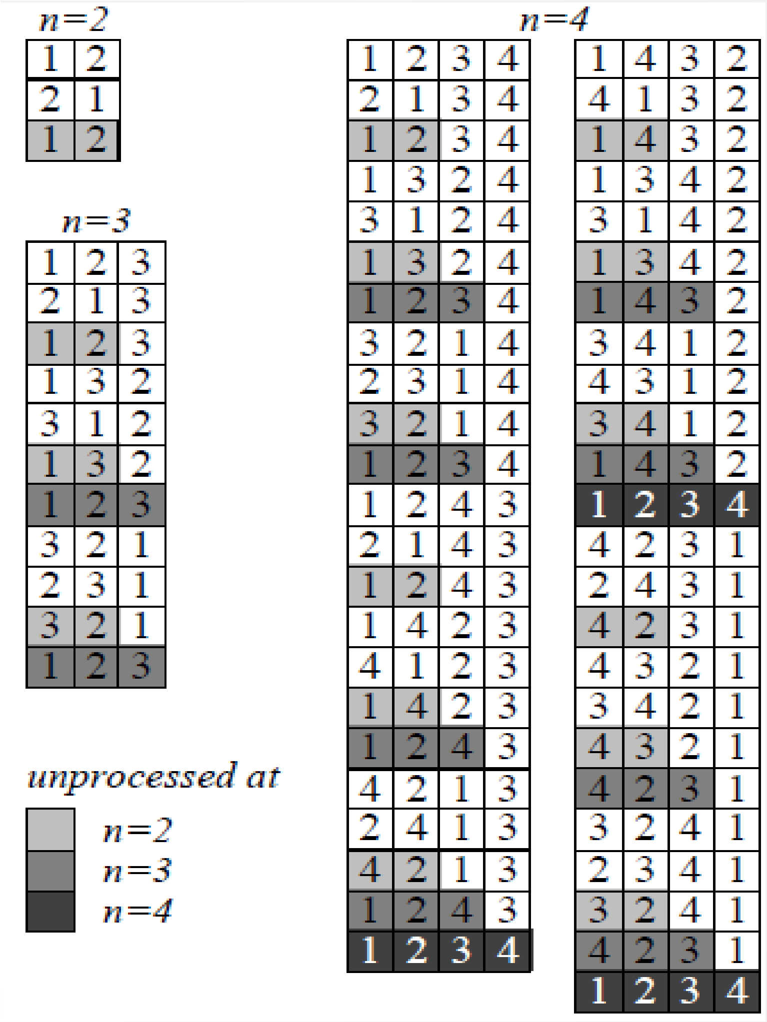
  ProcessPermutations(*A*, *k*-1);

  for *i* = *k*-1 downto 1 do

    Поменять A[*k*] и A[*i*]

    ProcessPermutations(*A*, *k*-1);

    Поменять A[*k*] и A[*i*]

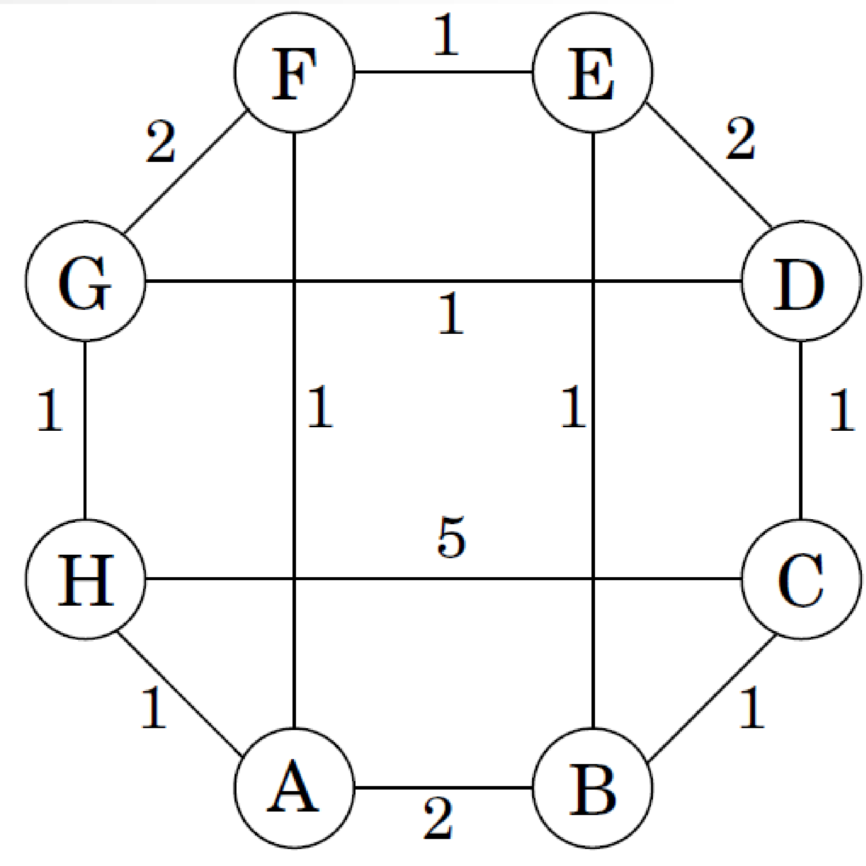


Решение ЗК

Для чего нужна процедура `Process()`?

1. Надо проверить корректность – представляет ли текущая перестановка допустимый (гамильтонов) цикл на заданном графе.
2. Если перестановка представляет корректный цикл, посчитать его вес и сравнить с весом цикла-*рекордсмена*.

Решение ЗК



Пример:

- Сгенерировать 7! перестановок, зафиксировав вершину A в качестве начальной.
- Перестановка ‘aBCDEFGH’ допустима, её вес = 11.
- Перестановка ‘aBCDEFHG’ не допустима, т.к. вершины F и H не смежны на графе.
- Перестановка ‘aFEBCHGD’ не допустима (не представляет гамильтонов цикл), т.к. вершина D не смежна с вершиной A.

Метод ветвей и границ

Метод ветвей и границ

- Предназначен в первую очередь для решения NP-трудных оптимизационных задач.
- Сочетание двух операций:
  - ✓ Ветвление.
  - ✓ Оценка верхней/нижней границы и «отсечение» неперспективной ветви.

Метод ветвей и границ

<u>Ветвление</u>

- Всё множество допустимых решений S(x) последовательно *виртуально* делится на подмножества:

$$S = \bigcup S_i$$

- Подмножества рекурсивно делятся дальше:

$$S_i = \bigcup S_{i,j}$$

- Для разделения множества решений на подмножества обычно последовательно накладывают условия/ограничения, которым должны удовлетворять решения из заданного подмножества.

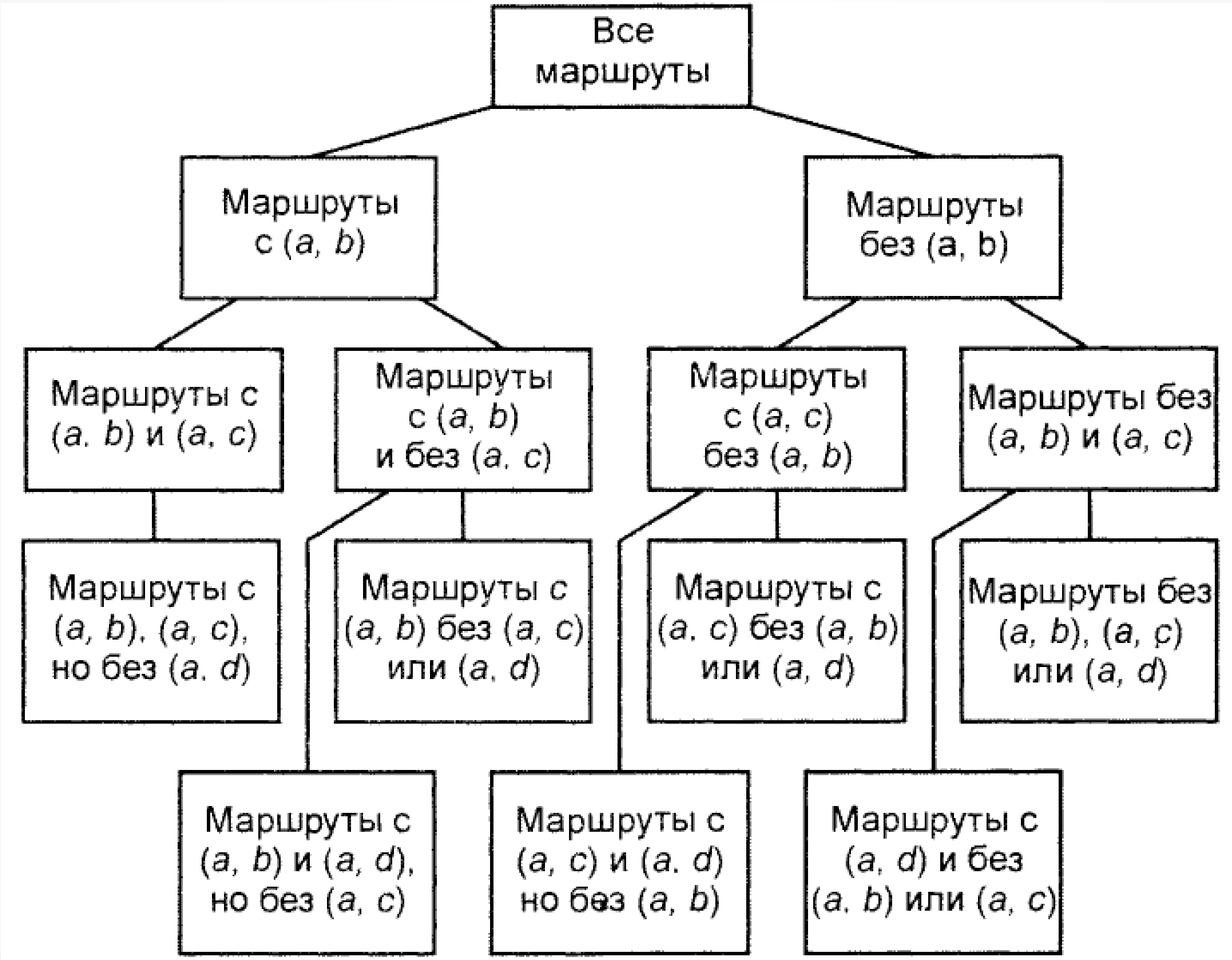
Метод ветвей и границ

Примеры ограничений для задачи коммивояжёра

- По принципу «содержит» / «не содержит» указанные дуги (алгоритм Литтла).

Выбрать дугу $e \in E$ и разделить S(x) на два подмножества: $S_{\{e\}}(x) = \{Z: e \in Z\}$ и $S_{\{\bar{e}\}}(x) = \{Z: e \notin Z\}$. Потом выбираем другую дугу и аналогичным образом делим $S_{\{e\}}(x)$ и $S_{\{\bar{e}\}}(x)$ на более мелкие подмножества.

Метод ветвей и границ



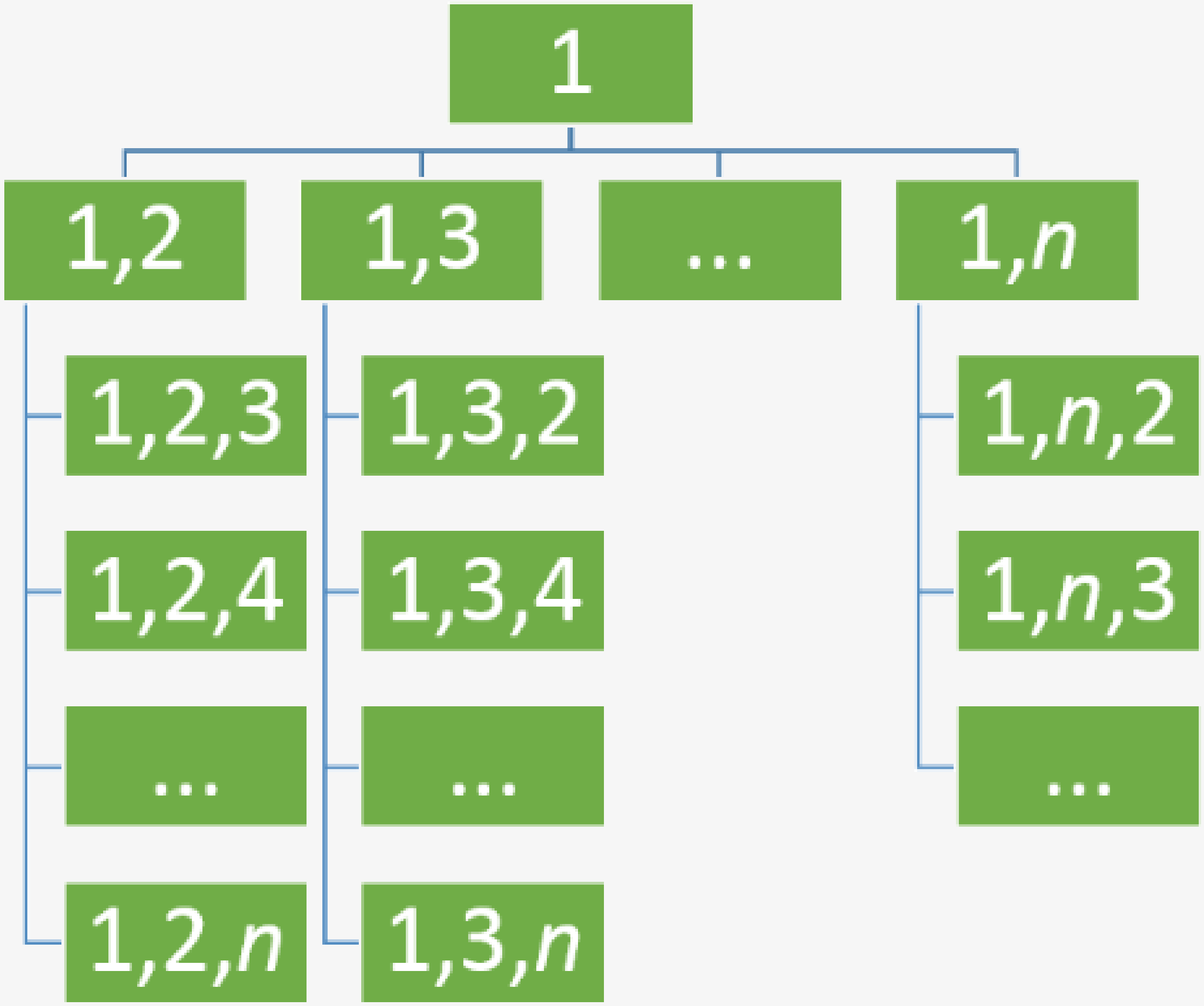
Метод ветвей и границ

Примеры ограничений для задачи коммивояжёра

- Частичный путь

Фиксируем фрагмент цикла — какие вершины цикл проходит последовательно друг за другом. Стартовая вершина всегда фиксирована: «1». Поэтому разделяем все возможные циклы на подмножества, содержащие фрагменты: «1,2», «1,3»,..., «1,*n*». Потом делим множество «1,2»: «1,2,3», «1,2,4», …, «1,2,*n*».

Метод ветвей и границ



Метод ветвей и границ

Принцип отсечения неперспективных ветвей

- Вычисляем оценку («границу») стоимости решений, входящих в анализируемое подмножество. Для задач минимизации вычисляем оценку снизу, для максимизации — оценку сверху.
- Сравниваем оценку со стоимостью рекордного (наилучшего на данный момент) решения. Если оценка хуже (для минимизации — выше, для максимизации — меньше), чем рекордное решение — данное подмножество далее не анализируем.

Метод ветвей и границ

Важно иметь в виду: на вычисление оценки также тратится время, и важно соблюдать баланс между качеством оценки и сложностью её вычисления.

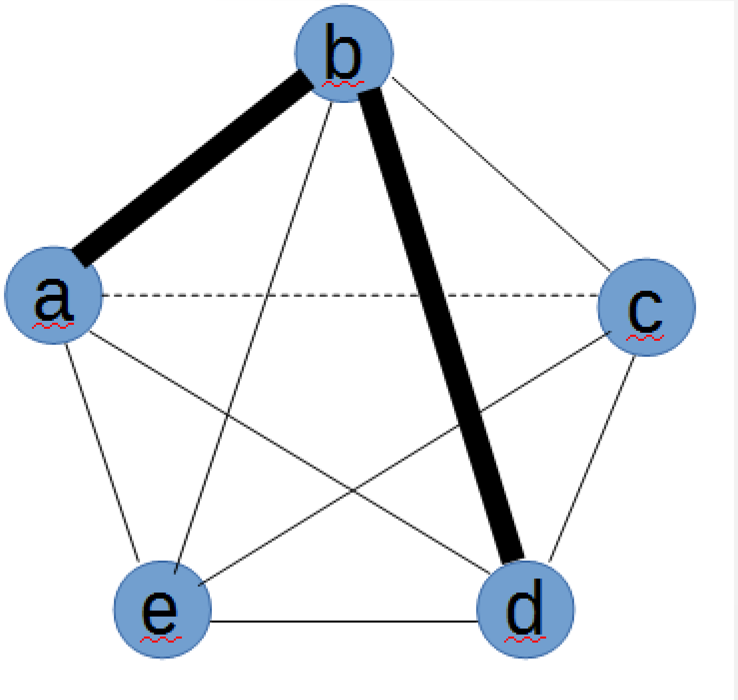
Возможный вариант: применять разные оценки на разных уровнях «дерева перебора»: на верхних уровнях более сложные, но более точные, на низших уровнях — менее точные, но более быстрые (**подумайте — почему так?**).

Метод ветвей и границ

Варианты вычисления границ (оценка снизу) для задачи коммивояжёра:

(1) Выбрать минимальную стоимость дуги (ещё не включённой в цикл), умножить на количество ещё не пройденных вершин графа и прибавить стоимости дуг, уже включённых в цикл.

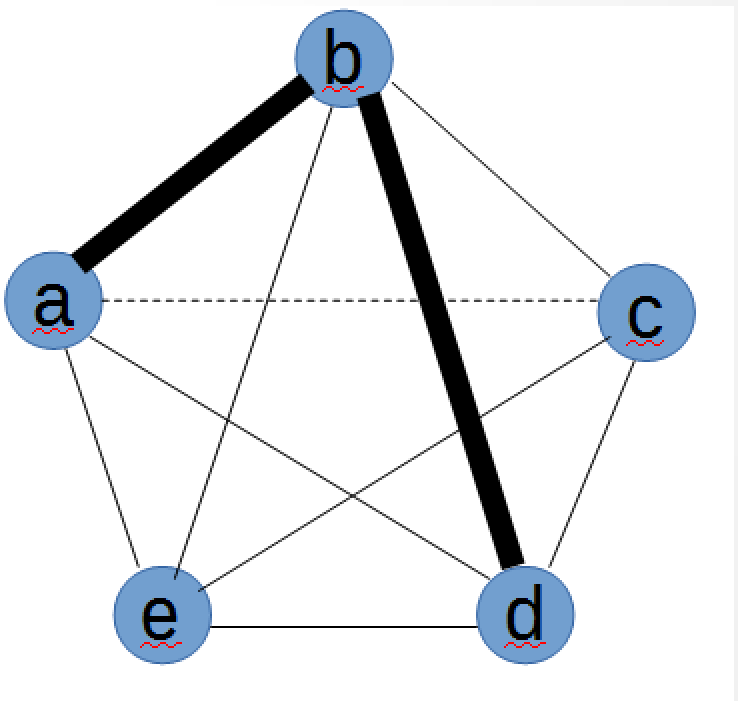
$$w' = w(a,b)+w(b,d)+3*w(a,c)$$



Метод ветвей и границ

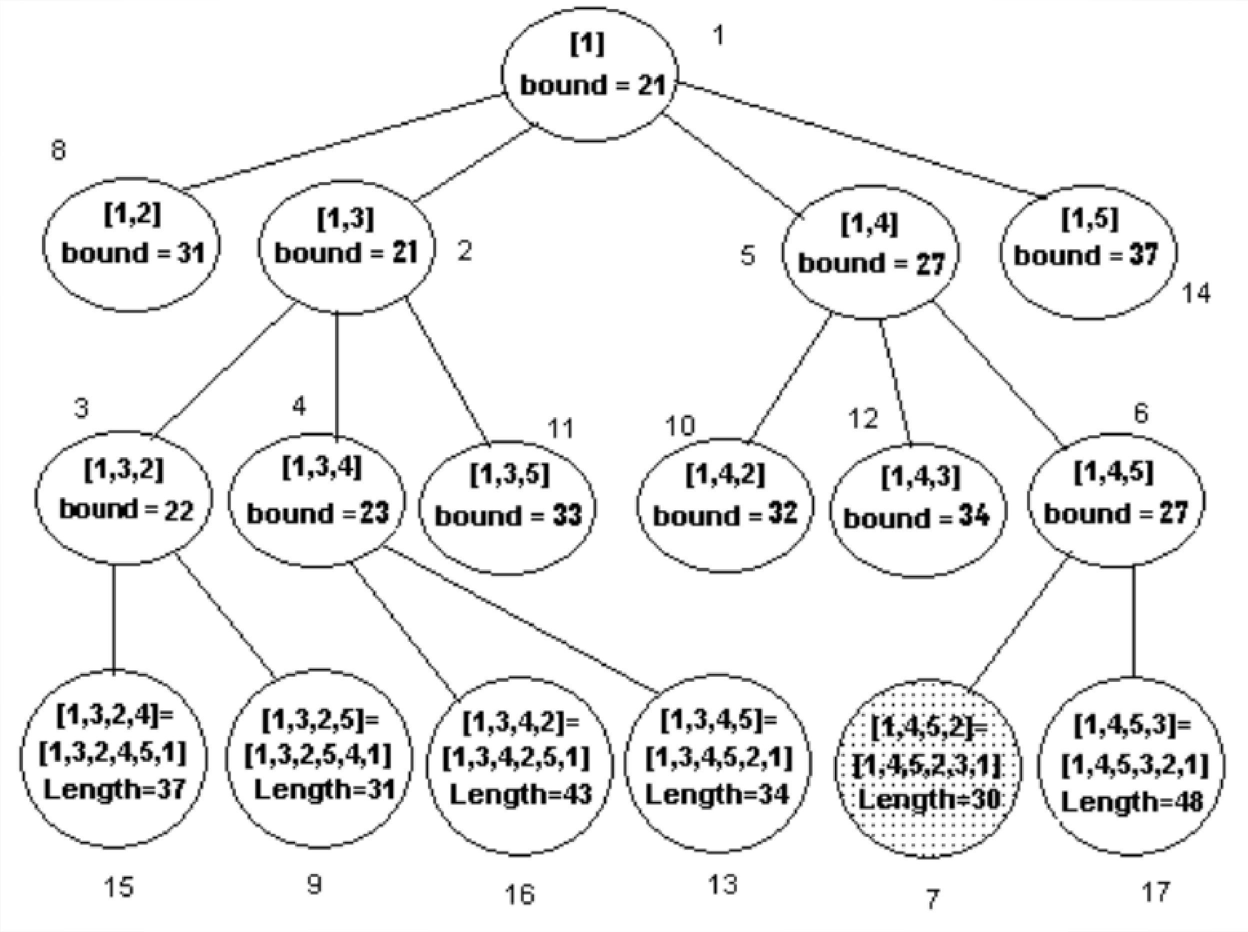
Варианты вычисления границ (оценка снизу) для задачи коммивояжёра:

(2) Для каждой вершины найти две инцидентные ей дуги наименьшей стоимости (обязательно включая дуги, уже вошедшие в цикл — если такие есть) и взять их среднее арифметическое. В качестве оценки использовать сумму этих средних



МВГ + «сначала наилучшие»

«Классический» МВГ = обход дерева решений в глубину



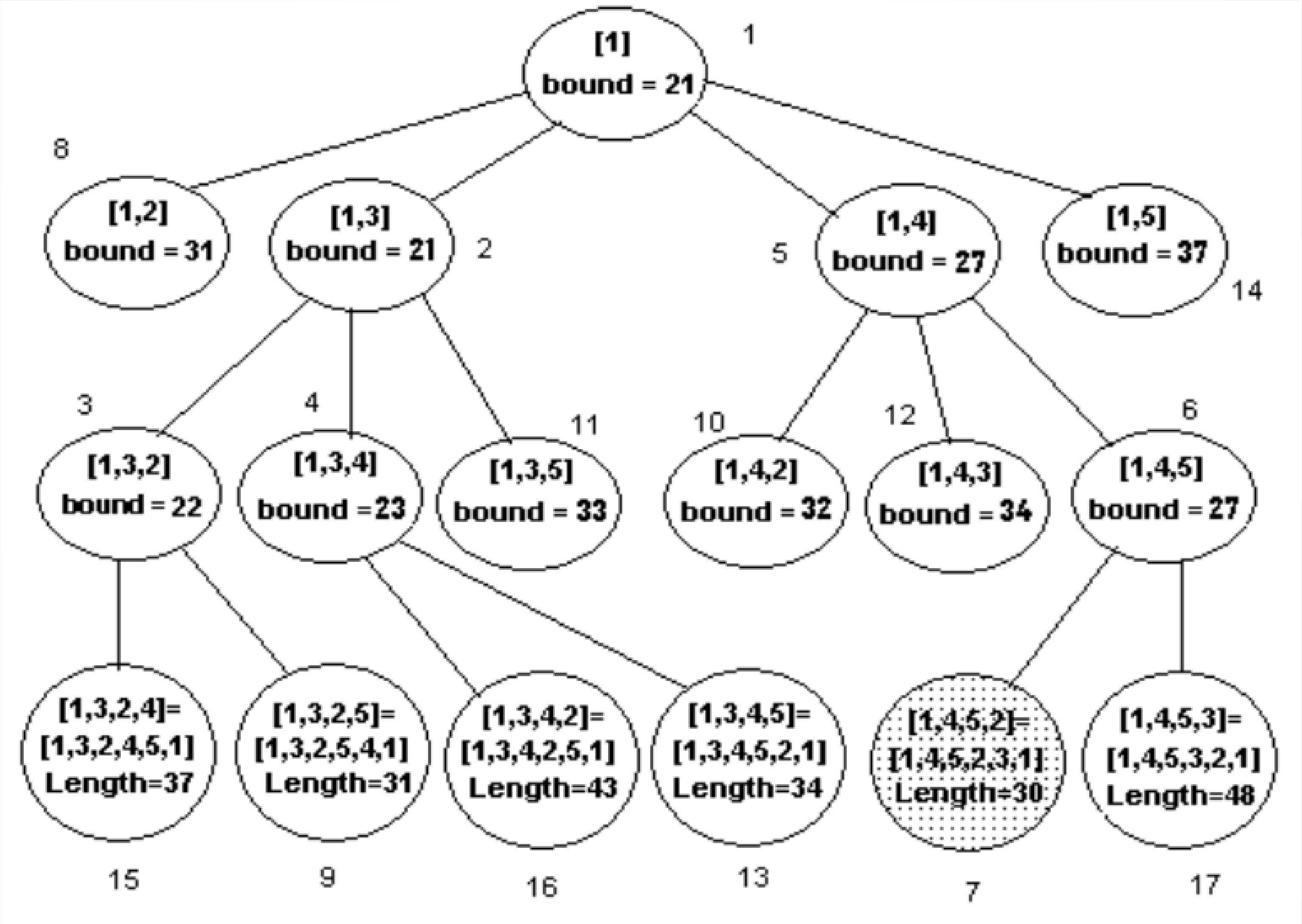
МВГ + «сначала наилучшие»

Применение эвристики «сначала наилучшие» («best-first»):

- Каждый «вариант» (подмножество решений) после расчёта границы сохраняем в очередь с приоритетами.
- Для ветвления берём не текущий (только что оцененный) вариант, а извлекаем из очереди наилучший (с минимальной/максимальной оценкой).

Метод содержит черты обхода дерева решений в ширину.

МВГ + «сначала наилучшие»



Метод ветвей и границ

Сложность метода ветвей и границ: в худшем случае – экспоненциальная, совпадает со сложностью полного перебора $O(n!)$.

Вопрос: как строить начальный цикл-рекордсмен?

1) Никак – инициализируем рекорд значением $+\infty$. 👎

2) Найти эвристическое (жадное?) решение. 👍