1.5em 0pt

Ananysis of Approximation Algorithms

L1: A Very Brief Introduction

杜睿

2024 年 10 月 25 日

Table of Contents

Overview

对于求解 NP 难（如组合优化）问题，有两条主线：

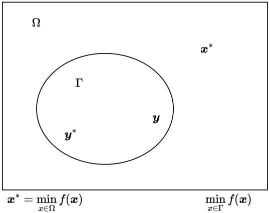
1. 如何定义优化问题（或对原问题进行转化），如何表达可行解与解空间，语义（表现型）和存储（基因型）可以有所不同。
2. 如何平衡局部搜索与跳坑策略，平衡开采与探索：
    - 如果开采不足，收敛性不好；
    - 如果探索不够，容易早熟，陷入局部最优解。

求解算法可分为确切算法和近似算法：

1. 确切算法保证给出最优解，但由于“组合爆炸”，仅可用于计算较小规模实例。
2. 近似算法或许有可能在短时间内，给出相当接近最优解的近似解，**对于狭义的近似算法特指又理论证明的近似算法**，进化算法等元启发算法求得近似解以实际效果为准，并不具备理论保证。

Design Techniques for Approximation Algorithms

- Greedy Strategy *(perturbation on the objective functions)*
- Restriction *(narrow down the feasible domain)*
  - Partition
  - Guillotine Cut
- Relaxation + Rounding *(enlarge the feasible domain to include infeasible solutions)*
  - Linear Programming
  - Primal-Dual Schema
  - Semidefinite Programming



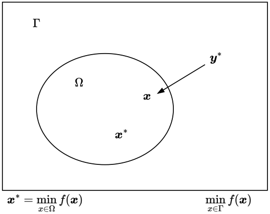


Table of Contents

Problem without Polynomial Time Algorithm

Vertex Cover

Given an undirected graph $G(V, E)$, find a subset $V' \subseteq V$ such that, "covers" all of the edges: for every edge $(u, v) \in E$, either $u \in V'$ or $v \in V'$ (or both). Furthermore, find a V' such that $|V'|$ is **minimum**.

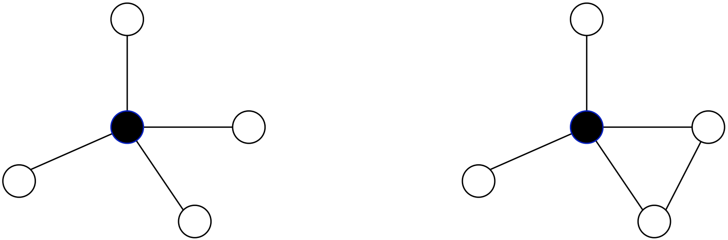
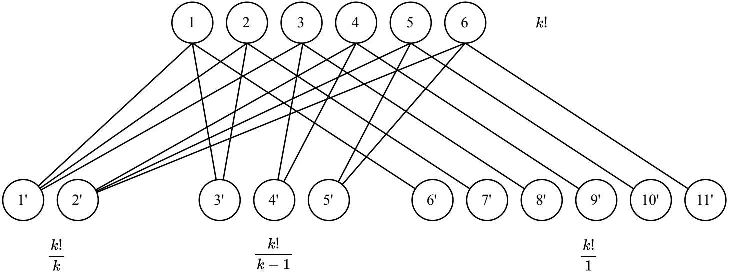


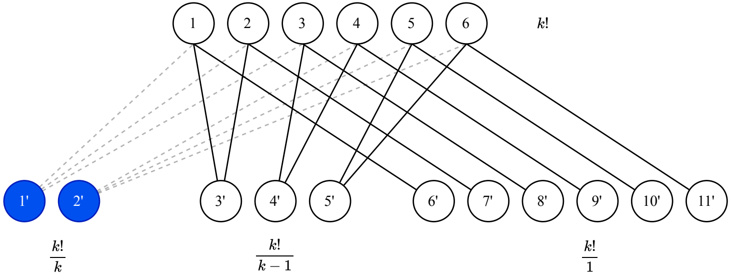
Figure: Feasible *vs*.Unfeasible

Pick the max degree continually



You may end up doing the wrong thing, if you have ties in terms of maximum degree.

Pick the max degree continually



Worst Case Analysis: $\log(k)$-Approximation

$$\frac{C}{C_{opt}} = \frac{k!(\frac{1}{k} + \frac{1}{k-1} + \cdots + \frac{1}{2} + \frac{1}{1})}{k!} \approx \log(k)$$

Pick random edges?

The edges we pick do not intersect with each other, or, the edges do not share vertices.

Algorithm 1 Approximation Algorithm for Vertex Cover

Input: Graph $G = (V, E)$
Output: The Number of Vertices Used $|C|$
$C \leftarrow \varnothing, E' \leftarrow \varnothing$
while $E' \neq \varnothing$ **do**
    Pick an edge $(u, v) \in E'$ arbitrarily
    $C \leftarrow C \cup \{u, v\}$
    Remove all edges incident on u or v from E'
end while
return $|C|$

Proof of 2-Approximation

- Let A denote the edges that are picked, then $C = 2|A|$.
- According to the definition, the optimal algorithm need to cover every edge, including all edges of A, then, $C_{opt} \geqslant |A|$, $\frac{C}{C_{opt}} \leqslant 2$

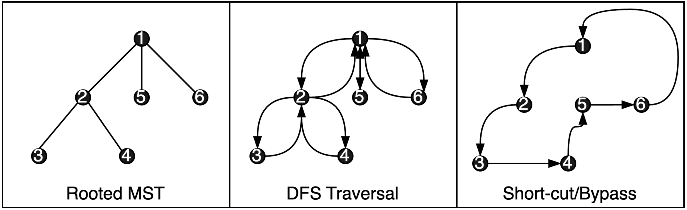
Table of Contents

Problem Description

给定 n 个城市，对这 n 个城市中的每两个城市来说，从一个城市到另一个城市所走的路程是已知的正实数（符合三角形三边关系定则），其中 n 是已知的正整数，$n \geqslant 3$。这 n 个城市的全排列共有 $n!$ 个。每一个这 n 个城市的全排列都恰好对应着一种走法：从全排列中的第一个城市走到第二个城市，$\cdots$，从全排列中的第 $n-1$ 个城市走到第 n 个城市，从全排列中的第 n 个城市回到第一个城市。要求给出一个这 n 个城市的全排列 σ，使得在 $n!$ 个全排列中，全排列 σ 对应的走法所走的路程是最短的(严格来讲，由于起点任意、顺逆时针等价，问题复杂度为 $\frac{(n-1)!}{2}$)。

2-Approximation

1. a: 定义 S 代表一系列边（允许重边），$c(S)$ 代表各边权重（长度）之和。
2. b: 定义 H_G^* 为无向多重图 G 上，长度最短的哈密尔顿回路（Hamiltonian Cycle），途中经过所有点且只经过一次。
3. c: 构造最小生成树 T，根据最小权生成树定义，$c\left(H_G^*\right) \geqslant c\left(H_G^* - e\right) \geqslant c(T)$。
4. d: 按深度优先搜索次序记录回路 C，下探一次，回溯一次，因此 $c(C) = 2 \times c(T)$。
5. e: 搭桥（short-cut/bypass）略过重复访问的点得到符合问题描述的新回路 C'（最后回到起点），例如，1，2，3，4，5，6. . . ，1。



Proof of 2-Approximation

- 由 e、三角形三条边关系定则，$c(C') \leqslant c(C)$；
- 由 c，$c\left(H_G^*\right) \geqslant c\left(H_G^* - e\right) \geqslant c(T)$；
- 由 d，$c(C) = 2 \times c(T)$;
- 故 $c(C') \leqslant 2c\left(H_G^*\right)$；
- 因此，该近似算法所得解，最多也不会超过最优解的 2 倍。

Proof of 2-Approximation

- 由 e、三角形三条边关系定则，$c\left(C'\right) \leqslant c\left(C\right)$；
- 由 c，$c\left(H_G^*\right) \geqslant c\left(H_G^* - e\right) \geqslant c\left(T\right)$；
- 由 d，$c\left(C\right) = 2 \times c\left(T\right)$;
- 故 $c\left(C'\right) \leqslant 2c\left(H_G^*\right)$；
- 因此，该近似算法所得解，最多也不会超过最优解的 2 倍。

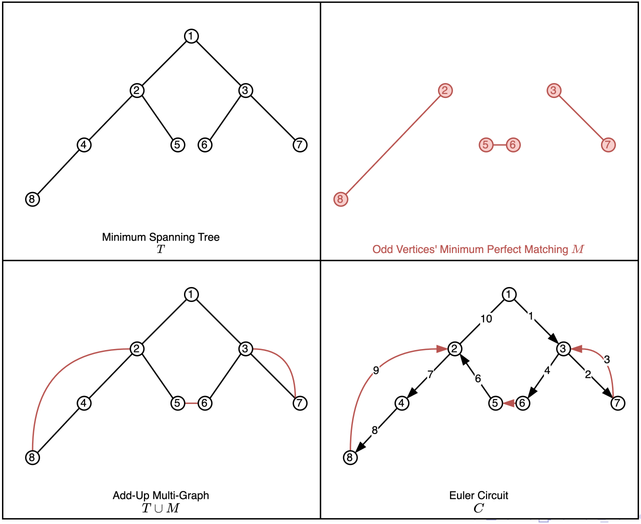
Improvement?

然后仍基于最小生成树，设法减小“每边下探一次，回溯一次”带来的额外开销，导出理论近似比为 1.5 的算法。期待一笔画、不重边地遍历所有顶点，可以将问题转换成“欧拉回路”问题。无向图存在欧拉回路的充要条件为：该图为连通图，且所有顶点度数均为偶数。倘若‘奇度数‘顶点为偶数个（证明见下)，那么可以通过将其两两匹配，为每一个顶点都“附赠”一个度，这样便可以满足“顶点度数均为偶数”条件。

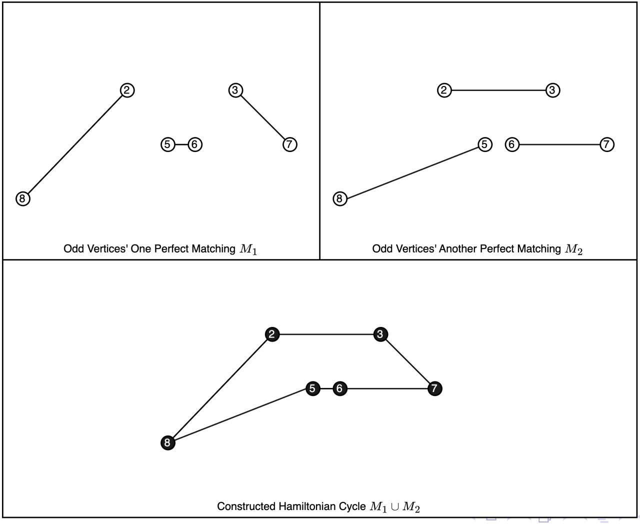
Christofides Algorithm

1. a: 定义 S 代表一系列边（允许重边），$c(S)$ 代表各边权重（长度）之和。
2. b: 定义 H_G^* 为无向多重图 G 上，长度最短的哈密尔顿回路（Hamiltonian Cycle），即途中经过所有点且只经过一次。
3. c: 定义假设 S 为无向多重图 G 上的导出子图，在 S 上长度最短的哈密尔顿回路记为 H_S^*。根据三角形三边关系定则易证，$c\left(H_S^*\right) \leqslant c\left(H_G^*\right)$。
4. d: 构造最小生成树 T，根据最小权生成树定义，$c\left(H_G^*\right) \geqslant c\left(H_G^* - e\right) \geqslant c(T)$。
5. e: 分离在 T 上度数为奇数的点，生成导出子图 S（根据握手定理，给定无向图 $G = (V, E)$，一条边贡献 2 度，故有 $\Sigma degG(v) = 2|E|$；除开度数为偶数的顶点所贡献的度数，推论可知，度数为奇数顶点数有偶数个）；
6. f: 构造 S 的最小权完美匹配 M，构造多重图 $G' = T \cup M$（此时每个顶点均为偶数度，故存在欧拉回路）；
7. g: 生成 G' 的欧拉回路 C，$c(C) = c(T) + c(M)$;
8. h: 搭桥（short-cut/bypass）略过重复访问的点（起点终点不删）得到符合问题描述的新回路 C'（最后回到起点）。

Demo


1
2
3
4
5
6
7
8
Minimum Spanning Tree
T
Odd Vertices' Minimum Perfect Matching M
Add-Up Multi-Graph
$T \cup M$
Euler Circuit
C
10
9

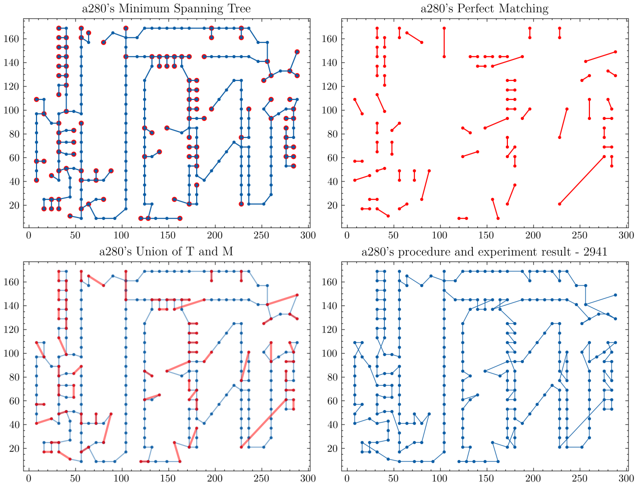
Demo


2
3
5
6
7
8
Odd Vertices' One Perfect Matching M_1
2
3
5
6
7
8
Odd Vertices' Another Perfect Matching M_2
2
3
5
6
7
8
Constructed Hamiltonian Cycle $M_1 \cup M_2$

Proof of 1.5 Approximation

- 由 e、三角形三边关系定则，$c(C') \leqslant c(C)$；
- 由 d，$c\left(H_G^*\right) \geqslant c\left(H_G^* - e\right) \geqslant c(T)$；
- 由 g，$c(C) = c(T) + c(M)$；
- 由 f、c，$c(M) + c(M) \leqslant c(M1) + c(M2) = c\left(H_S^*\right) \leqslant c\left(H_G^*\right)$，得 $c(M) \leqslant \frac{1}{2} c\left(H_G^*\right)$；
- 故 $c(C') \leqslant c(T) + c(M) \leqslant c\left(H_G^*\right) + \frac{1}{2} c\left(H_G^*\right)$；
- 即得证。

Proof of 1.5 Approximation



Extra Material

Benchmark		Proposed GIGA			WangLei			SimulatedAnnealing		
Name	L_{OPT}	$L_{\min}$	L_{avg}	t_{avg}	$L_{\min}$	L_{avg}	t_{avg}	$L_{\min}$	L_{avg}	t_{avg}
a280	2579	**2584**†	2593.30	312.50	2615	2653.30	380.50	2792	2890.40	43.76
berlin52	7542	**7542**⋆	7542.00	36.73	**7542**⋆	7542.00	4.79	**7542**⋆	7759.30	8.06
bier127	118282	120843	121648.60	97.18	**118326**†	119221.50	51.17	121173	124320.50	19.63
ch130	6110	6189	6198.00	93.35	**6115**†	6131.90	43.74	6355	6548.00	19.89
ch150	6528	6588	6588.00	112.96	**6554**†	6582.50	65.98	6938	7069.70	22.98
d198	15780	15831	15888.30	194.36	**15818**†	15860.00	141.29	16211	16464.80	30.36
d493	35002	**35544**†	35560.27	1239.37	35670	35838.82	3028.20	39580	40399.09	110.69
d657	48912	**49852**†	49900.80	2429.40	50101	50247.40	7525.27	61152	62870.10	157.57
eil51	426	435	435.40	254.25	**426**⋆	427.00	31.93	429	435.60	48.15
eil76	538	546	546.00	374.14	**542**†	545.10	89.43	556	560.20	75.42
eil101	629	639	641.20	473.52	**633**†	636.30	162.24	656	665.70	92.91
fl417	11861	11962	11977.00	947.40	**11899**†	11933.20	1360.14	13088	13604.10	90.50
gil262	2378	2394	2402.50	459.81	**2391**†	2411.30	519.80	2541	2628.20	71.41
p654	34643	**34647**†	34839.70	2067.31	34806	34959.60	5173.19	42302	44315.60	162.21
pcb442	50778	51338	51338.70	949.10	52128	52553.30	2043.72	57294	59100.20	99.86
pr76	108159	109043	109043.00	253.17	**108159**⋆	108257.90	60.67	109696	111023.00	52.94
pr107	44303	**44303**⋆	44497.70	364.35	**44303**⋆	44330.50	112.80	45179	46623.40	74.77
pr124	59030	**59030**⋆	59030.00	452.88	**59030**⋆	59034.60	160.70	60073	61349.70	87.81
pr136	96772	**96772**⋆	96781.10	520.55	96795	96985.20	288.84	100677	102998.60	95.56
pr144	58537	58763	59162.80	603.83	**58537**⋆	58642.40	263.35	59127	60989.10	102.01
pr152	73682	73880	73880.00	597.60	**73682**⋆	73737.80	286.54	75208	76857.00	110.15

† 代表在当前评价指标上优于其他算法；⋆ 代表在该测试用例上找到最优解。

Bibliography

[1] R. Du, "My implementation." https://github.com/DURUII/Homework-Algorithm-TSPLIB95, 2024.

[2] E. Demaine and S. Devadas, "6.046j design and analysis of algorithms, spring 2015." MIT OpenCourseWare, ocw.mit.edu/6-046J-spring2015, 2015.

[3] J. O. Nelson, "Efficient algorithms and intractable problems, spring 2020," 2020. Lecture notes from CS170.

[4] D.-Z. Du, K.-I. Ko, and X. Hu, *Design and analysis of approximation algorithms*, vol. 62. Springer Science & Business Media, 2011.

[5] V. V. Vazirani, "Approximation algorithms," 2001.