# CMPT307: Approximation Algorithms

Week 13-3

**Xian Qiu**

Simon Fraser University

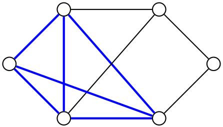
xianq@sfu.ca

# Outline

- ▷ more NP-complete problems
- ▷ approximation algorithms

# Clique

a clique of an undirected graph $G = (V, E)$ is a subset $V' \subseteq V$ that induces a complete graph



a clique of size $4$

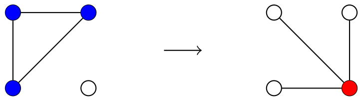
Clique: does $G$ contain a clique of size at least $k$?

Vertex-Cover $\leq_P$ Clique

# $\textsc{Clique}$ is NP-complete

$\textsc{Vertex-Cover}$: does $G$ have a vertex cover of size at most $k$?

create its complement graph $\bar{G} = (V, \bar{E})$

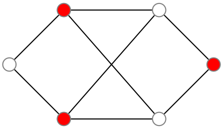


Claim. $\textsc{Clique}$ has size $k \Leftrightarrow$ $\textsc{Vertex-Cover}$ has size $n - k$.

- $U$ is a clique of $G \Leftrightarrow V \backslash U$ is a vertex cover of $\bar{G}$

# Independent-Set

an independent set of an undirected $G = (V, E)$ is a subset $V' \subseteq V$ such that no two vertices are adjacent



an independent set of size $3$

Independent-Set: does $G$ have an independent set of size $\geq k$?

Clique $\leq_P$ Independent-Set

consider $\bar{G}$

# More NP-complete Problems

## SUBSET-SUM

- given: $a_1, \ldots, a_n \in \mathbb{Z}$ and $A \in \mathbb{Z}$
- question: does there exist $S \subseteq \{1, \ldots, n\}$ s.t. $\sum_{i \in S} a_i = A$?

## PARTITION

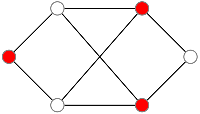
- given: $a_1, \ldots, a_n \in \mathbb{Z}$
- question: does there exist $S \subseteq \{1, \ldots, n\}$ s.t. $\sum_{i \in S} a_i = \frac{1}{2} \sum_{i=1}^{n} a_i$?

## 3-PARTITION

- given: integers $a_1, \ldots, a_{3n}$, with $\frac{A}{4} < a_i < \frac{A}{2}$, where $A = \frac{1}{n} \sum_{i=1}^{3n} a_i$
- question: can they be partitioned into $n$ triplets s.t. $\sum_{i \in S_j} a_i = A$?

# Approximating Vertex Cover

given $G = (V, E)$, find a vertex cover $C$ of minimum size



- no polynomial time exact algorithm unless $\mathcal{P} = \mathcal{NP}$
- try to approximate OPT as close as possible (in poly-time)

## Approximation algorithm

$\mathbb{A}$ is an $\alpha$-approximation algorithm $(\alpha > 1)$ for vertex cover if

- $\mathbb{A}$ runs in polynomial time
- $\mathbb{A}(I)$ outputs a (feasible) vertex cover for any instance $I$
- $\mathbb{A}(I)/\text{OPT} \leq \alpha$, for all $I$

# Approximating Vertex Cover

APPROX-VC($G$)

find a maximal matching $M$ of $G$;
**return** $V(M)$;

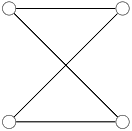
- $V(M)$ is a vertex cover and the algorithm is poly-time
- approximation ratio $\alpha = ?$

$$\begin{array}{l} \text{ALG} = 2\,|M| \\ \text{OPT} \geq |M| \end{array} \quad \Rightarrow \quad \frac{\text{ALG}}{\text{OPT}} \leq 2$$

- APPROX-VC is a 2-approximation algorithm for vertex cover

# Tightness of Analysis

- is the analysis good enough?
  say, is the algorithm indeed $1.9$-approximation?
- tightness of analysis: present an instance which attains $\alpha$-approximation



OPT $= 2$

ALG $= 4$

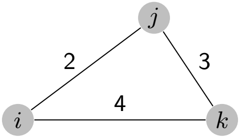
can we do better?

- no $\alpha < 1.3606$ approx. unless $\mathcal{P} = \mathcal{NP}$ — Dinur & Safra, 2005
- no $\alpha < 2$ approx. unless UGC fails — Khot & Regev, 2008

# Metric TSP

TSP (undirected): given an undirected complete graph $G = (V, E)$ and a cost function $c : E \to \mathbb{R}^+$, find a "tour" of minimum cost

metric TSP: $c$ is a metric, *i.e.*, $c_{ik} \le c_{ij} + c_{jk}$

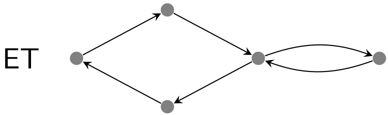



metric TSP is NP-hard

# Euler Tour

**Definition**

An Euler tour is a closed walk in a graph (parallel edges are also allowed) that traverses each edge exactly once.



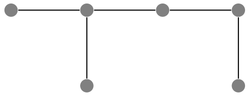

**Theorem**

For undirected graph, ET exists iff each node has even degree.

- can be found in $O(m)$ if it exists (*cf.* Q2 of H6)

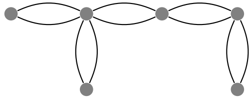
# Double-Tree Algorithm

1. compute a minimum spanning tree (MST)



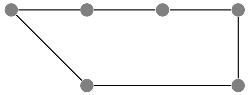
MST $\leq$ OPT

2. double the MST = Euler tour (ET)



ET $\leq 2$OPT

3. short cutting (use $\Delta$-inequality)

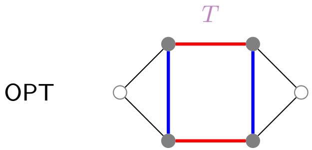


ALG $\leq 2$OPT

# A Better Way of Constructing ET

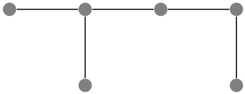
- only need to add an edge odd-degree nodes (of MST)
- add minimum cost perfect matching $M$ (on odd-degree nodes)
- does perfect matching always exist? $\sum_v d(v) = 2m$

Lemma. OPT $\geq 2c(M)$.



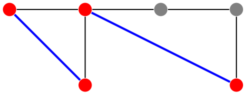

- walk along OPT and short cut on $v(M)$, clearly OPT$\geq c(T)$
- $T$ can be decomposed into perfect matchings $M_1$ and $M_2$
- OPT$\geq c(T) = c(M_1) + c(M_2) \geq 2c(M)$

1. compute a minimum spanning tree (MST)



MST $\leq$ OPT

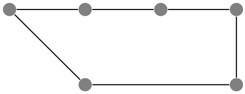
2. compute a min-cost perfect matching $M$ on odd nodes and find an Euler tour (ET)



$c(M) \leq \frac{1}{2}$OPT

ET $\leq \frac{3}{2}$OPT

3. short cutting (use $\Delta$-inequality)



ALG $\leq \frac{3}{2}$OPT