Approximation Algorithms

An introduction to Approximation Algorithms

Presented By Iman Sadeghi

Overview

- Introduction
- Performance ratios
- The vertex-cover problem
- Traveling salesman problem
- Set cover problem

Introduction

- There are many important NP-Complete problems
  - There is no fast solution !
- But we want the answer …
  - If the input is small use backtrack.
  - Isolate the problem into P-problems !
  - Find the **Near-Optimal** solution in polynomial time.

Performance ratios

- We are going to find a Near-Optimal solution for a given problem.
- We assume two hypothesis :
  - Each potential solution has a positive cost.
  - The problem may be either a maximization or a minimization problem on the cost.

Performance ratios …

- If for any input of size n, the cost C of the solution produced by the algorithm is within a factor of $\rho(n)$ of the cost C* of an optimal solution:

$$\text{Max}\left(C/C^* , C^*/C \right) \le \rho(n)$$

- We call this algorithm as an $\rho(n)$-approximation algorithm.

Performance ratios …

- In Maximization problems:

  $0<C\leq C^*$, $\rho(n) = C^*/C$
- In Minimization Problems:

  $0<C^*\leq C$, $\rho(n) = C/C^*$
  - $\rho(n)$ is never less than 1.
  - A 1-approximation algorithm is the optimal solution.
  - The goal is to find a polynomial-time approximation algorithm with small constant approximation ratios.

Approximation scheme

- **Approximation scheme** is an approximation algorithm that takes Є>0 as an input such that for any fixed Є>0 the scheme is (1+Є)-approximation algorithm.
- **Polynomial-time approximation scheme** is such algorithm that runs in time polynomial in the size of input.
  - As the Є decreases the running time of the algorithm can increase rapidly:
    - For example it might be $O(n^{2/\epsilon})$

Approximation scheme

- We have **Fully Polynomial-time approximation scheme** when its running time is polynomial not only in n but also in $^1/_\epsilon$
    - For example it could be $O((^1/_\epsilon)^3 n^2)$

Some examples:

- Vertex cover problem.
- Traveling salesman problem.
- Set cover problem.

The vertex-cover problem

- A vertex-cover of an undirected graph G is a subset of its vertices such that it includes at least one end of each edge.
- The problem is to find minimum size of vertex-cover of the given graph.
- This problem is an NP-Complete problem.

The vertex-cover problem …

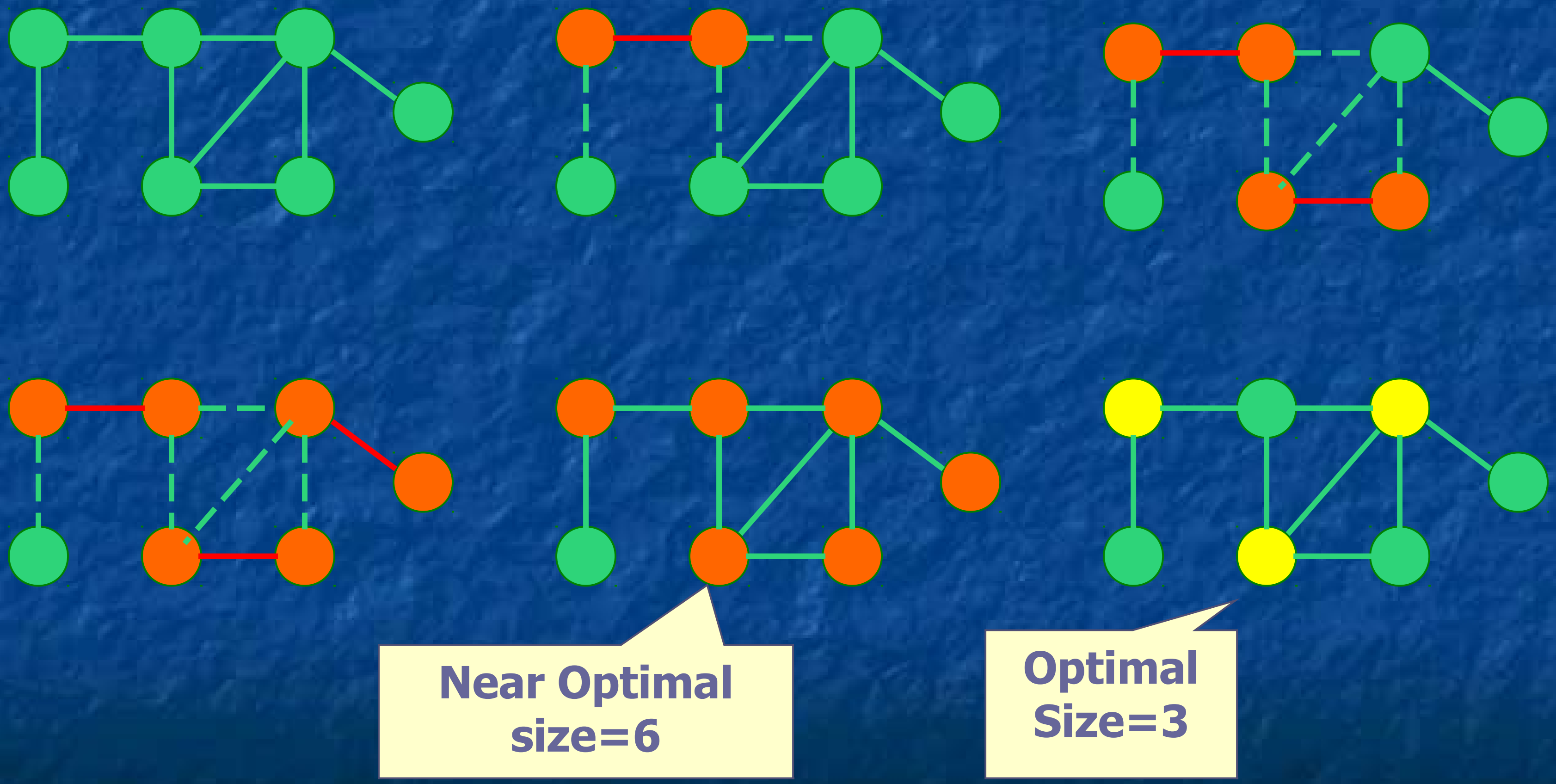
- Finding the optimal solution is hard (its NP!) but finding a near-optimal solution is easy.
- There is an 2-approximation algorithm:
  - It returns a vertex-cover not more than twice of the size optimal solution.

The vertex-cover problem …

APPROX-VERTEX-COVER(G)

```
C ← Ø
E′ ← E[G]
while E′ ≠ Ø
        do let (u, v) be an arbitrary edge of E′
                C ← C U {u, v}
                remove every edge in E′ incident on u or v
return C
```

The vertex-cover problem …



The vertex-cover problem …

- This is a polynomial-time 2-aproximation algorithm. (Why?)
  - Because:
    - **APPROX-VERTEX-COVER is O(V+E)**
    - **$|C^*| \geq |A|$**
      $|C| = 2|A|$
      $|C| \leq 2|C^*|$

Optimal

Selected Edges

Selected Vertices

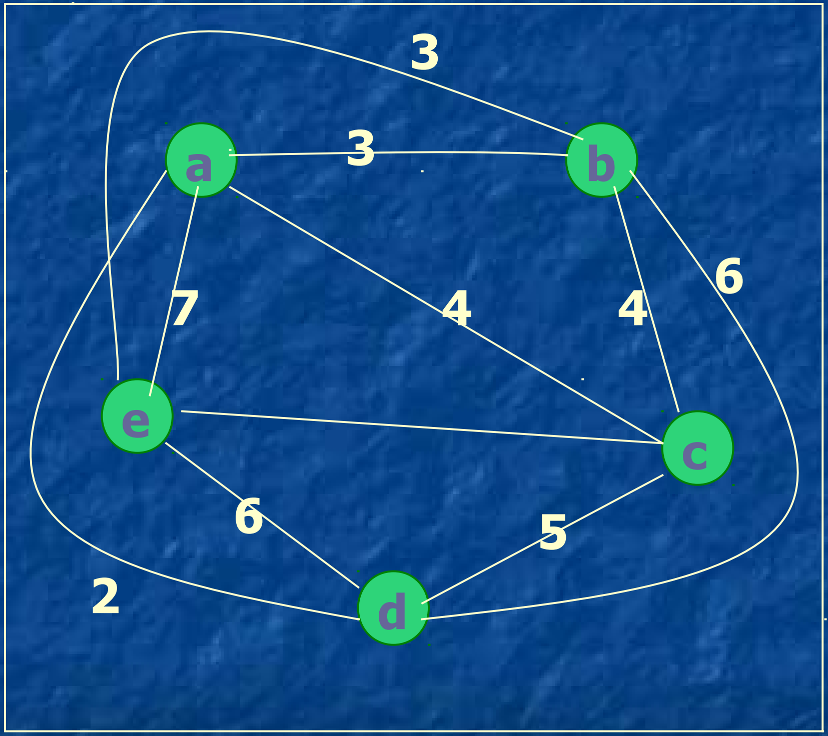
Traveling salesman problem

- Given an undirected weighted Graph G we are to find a minimum cost Hamiltonian cycle.
- Satisfying triangle inequality or not this problem is NP-Complete.
  - We can solve Hamiltonian path.

Traveling salesman problem

- Exact exponential solution:
  - Branch and bound
    - Lower bound:
      (sum of two lower degree of vertices)/2

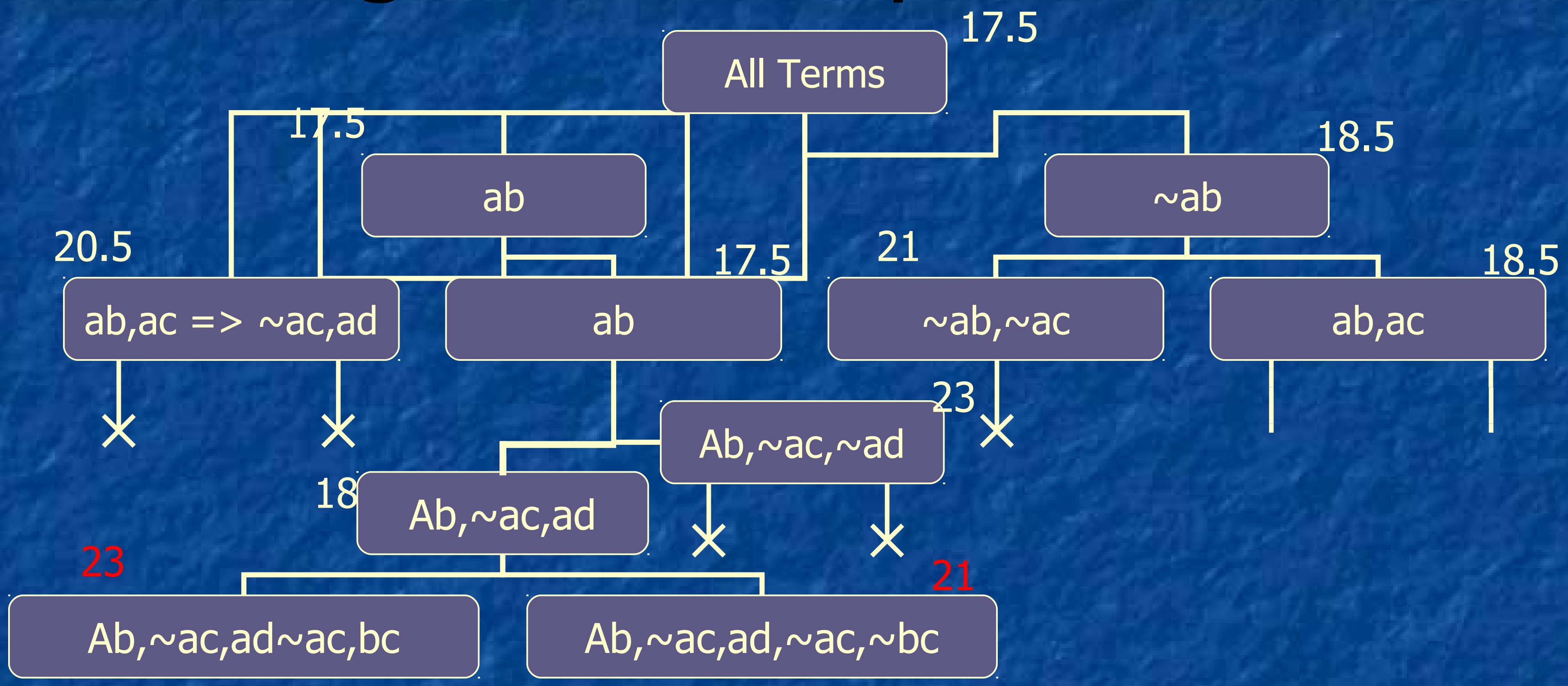
Traveling salesman problem



A: 2+3
B: 3+3
C: 4+4
D: 2+5
E: 3+6
= 35

Bound: 17,5

Traveling salesman problem



Traveling salesman problem

- Near Optimal solution
  - Faster
  - More easy to impliment.

Traveling salesman problem with triangle inequality.

APPROX-TSP-TOUR(G, c)

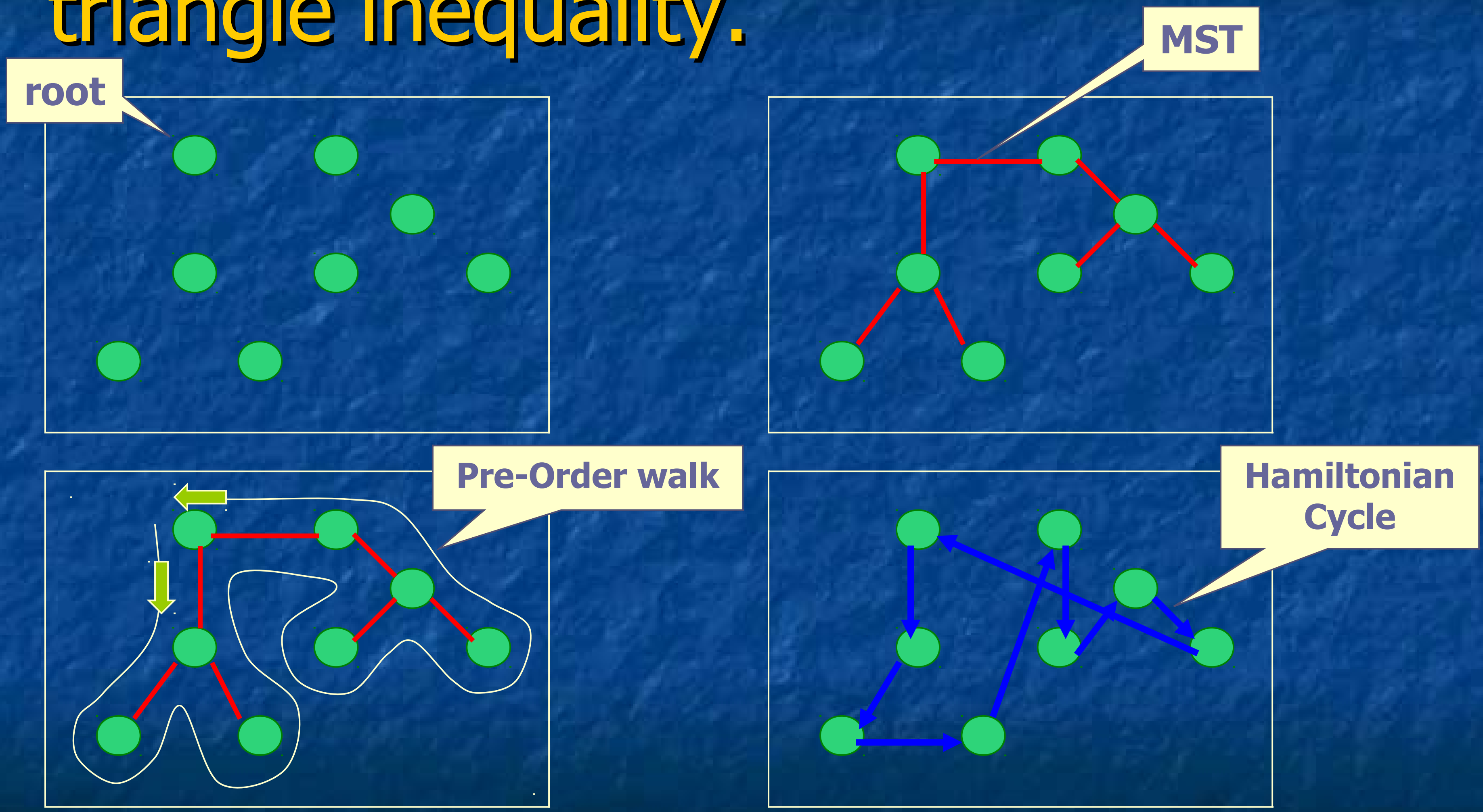
1 select a vertex **r** ϵ V[**G**] to be root.

2 compute a **MST** for **G** from root r using Prim Alg.

3 **L**=list of vertices in preorder walk of that **MST**.

4 **return** the Hamiltonian cycle **H** in the order **L**.

Traveling salesman problem with triangle inequality.



Traveling salesman problem

- This is polynomial-time 2-approximation algorithm. (Why?)
  - Because:
    - APPROX-TSP-TOUR is $O(V^2)$
    - $C(MST) \leq C(H^*)$ Optimal

      $C(W)=2C(MST)$ Pre-order

      $C(W) \leq 2C(H^*)$

      $C(H) \leq C(W)$ Solution

      $C(H) \leq 2C(H^*)$

Traveling salesman problem In General

- Theorem:

  If $P \neq NP$, then for any constant $\rho \geq 1$, there is no polynomial time ρ-approximation algorithm.

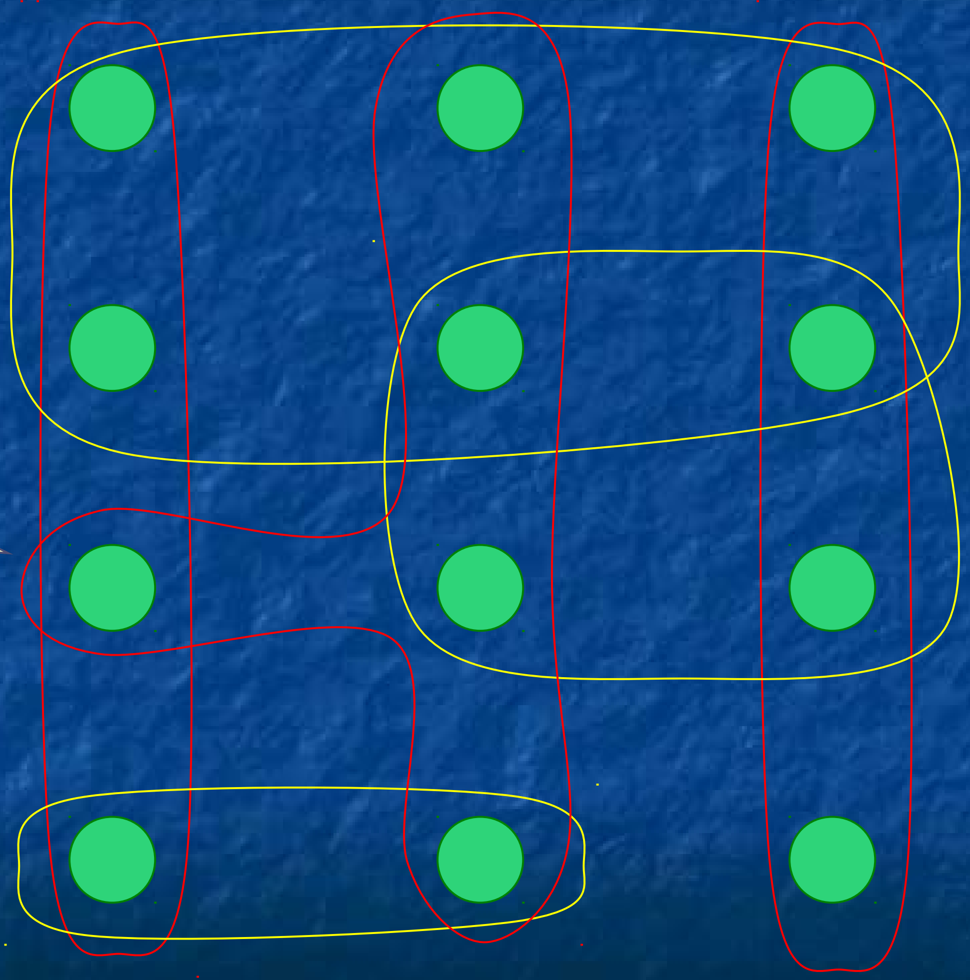
- $c(u,w) = \{uEw\ ?\ 1 : \rho|V|+1\}$

  $$\underbrace{\rho|V|+1}_{\text{Selected edge not in E}} + \underbrace{|V|-1}_{\text{Rest of edges}} > \rho|V|$$

The set-Cover

- Generalization of vertex-cover problem.
- We have given (X,F) :
  - X : a finite set of elements.
  - F : family of subsets of X such that every element of X belongs to at least one subset in F.
  - Solution C : subset of F that Includes all the members of X.

The set-Cover …

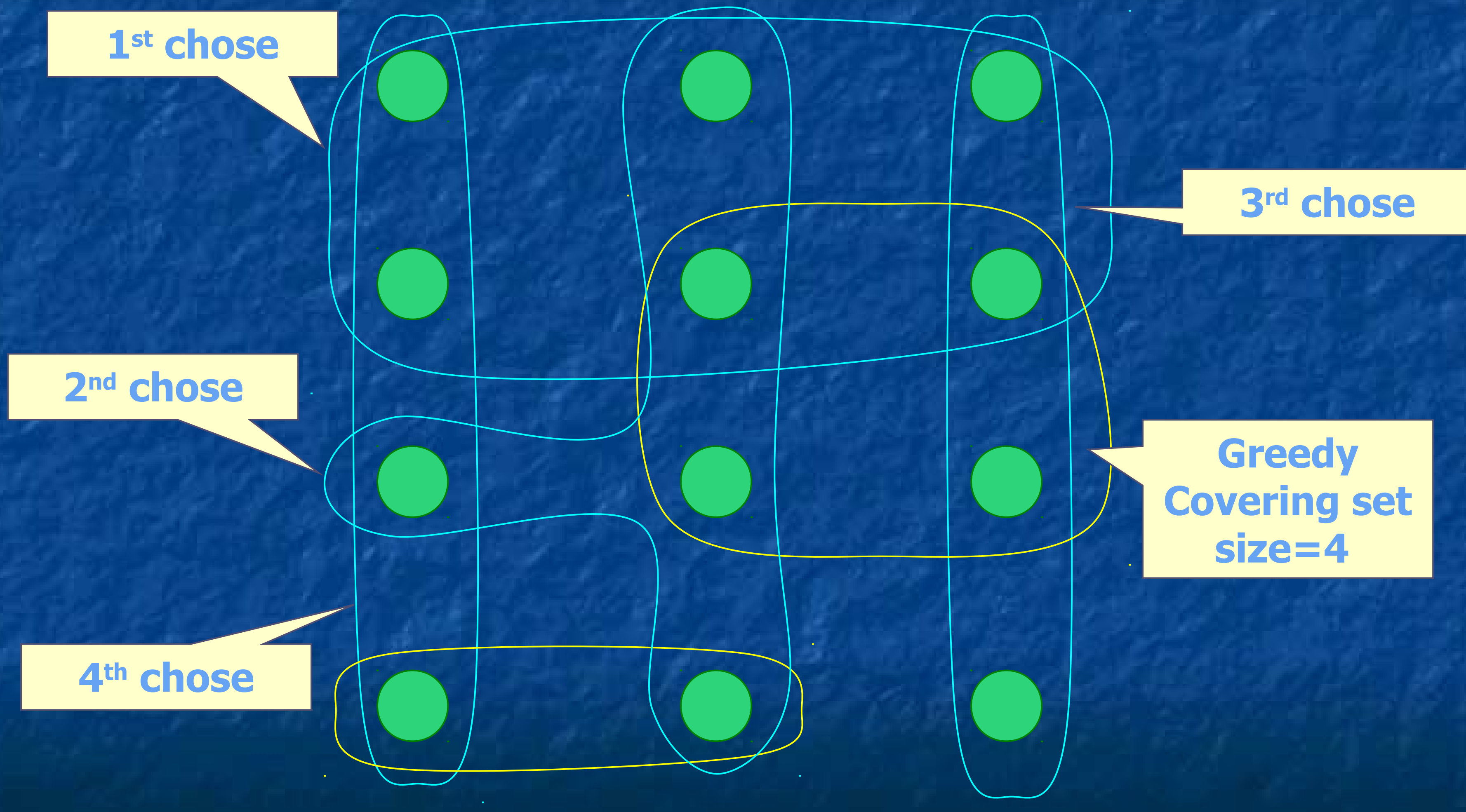


Minimal Covering set size=3

The set-Cover …

```
GREEDY-SET-COVER(X,F)
M ← X
C ← Ø
while M ≠ Ø do
        select an S∈F that maximizes |S ∩ M|
                M ← M – S
                C ← C ∪ {S}
return C
```

The set-Cover …



The set-Cover …

- This greedy algorithm is polynomial-time $\rho(n)$-approximation algorithm
  - **$\rho(n)=H(\max\{|S| : S \in F\})$**
  - **$H_d = \sum_{i=1}^{d}$**
- The proof is beyond of scope of this presentation.

Any Question?

Thank you
for your attendance
and attention.