Introduction to Algorithms

Meng-Tsung Tsai

12/19/2019

Reminder

10:20, Dec 24	23:59, Dec 27	13:30-17:30, Dec 28
written assignement #3	programming assignment #3	programing quiz #2

Bin Packing Problem

Problem Definition

Given a set S of n real numbers $a_1, a_2, ..., a_n \in (0, 1]$, partition them into subsets $S_1, S_2, ..., S_k$ so that

(1) $S_1 \cup S_2 \cup ... \cup S_k = S$,

(2) $\sum_{a \in S_i} a \leq 1$ for each $i \in [1, k]$, and

(3) k is minimized.

2-approximation algorithm

```
First-Fit (S){
    B₁ = B₂ = ... = Bₙ = 0;
    foreach ( a ∈ S ){
        for ( i = 1; i ≤ n; ++i){
            if(Bᵢ + a ≤ 1){
                Bᵢ += a; break;
            }
        }
    }
}
```

2-approximation algorithm

First-Fit (S){
    $B_1 = B_2 = \ldots = B_n = 0$;
    foreach ($a \in S$){
        for ($i = 1$; $i \leq n$; ++i){
            if($B_i + a \leq 1$){
                B_i += a; break;
            }
        }
    }
}

First-Fit gives a 2-approximation because one cannot find two non-zero B_i's that have value $\leq 1/2$.

2-approximation algorithm

```
First-Fit (S){
    B1 = B2 = ... = Bn = 0;
    foreach ( a ∈ S ){
        for ( i = 1; i ≤ n; ++i){
            if(Bi + a ≤ 1){
                Bi += a; break;
            }
        }
    }
}
```

It is shown that First-Fit gives a 1.7-approximation.
You may find a proof in
"First Fit bin packing: A tight analysis," Dósa and Sgall.

2-approximation algorithm

```
First-Fit (S){
    B1 = B2 = ... = Bn = 0;
    foreach ( a ∈ S ){
        for ( i = 1; i ≤ n; ++i){
            if(Bi + a ≤ 1){
                Bi += a; break;
            }
        }
    }
}
```

Can you implement First-Fit or an alternative in O(n log n) time?

NP-hardness to have (3/2-ε)-approxiamtion

It is known that Subset-Sum is NP-hard.

Given an integer W and a set S of n integers, decide whether there exists a subset of S whose elements sum to W.

⇒ Asking whether two bin is enough is NP-hard.

Why?

⇒ Having an (3/2-ε)-approximation is NP-hard.

Why?

Traveling Salesman Problem (general version)

Can we approximate TS-tour without assuming that the input satisfies the triangle inequality?

Claim. TS-tour cannot be approximated to within any constant factor in polynomial time unless P = NP.

(TS-tour $\notin$ APX, the class of NP optimization problems that can be approximated to within a constant factor in polynomial time.)

Proof.

Assume for contradiction that there exists an algorithm A that can approximate TS-tour to within a constant factor ρ in polynomial time.

Can we approximate TS-tour without assuming that the input satisfies the triangle inequality?

Claim. TS-tour cannot be approximated to within any constant factor in polynomial time unless P = NP.

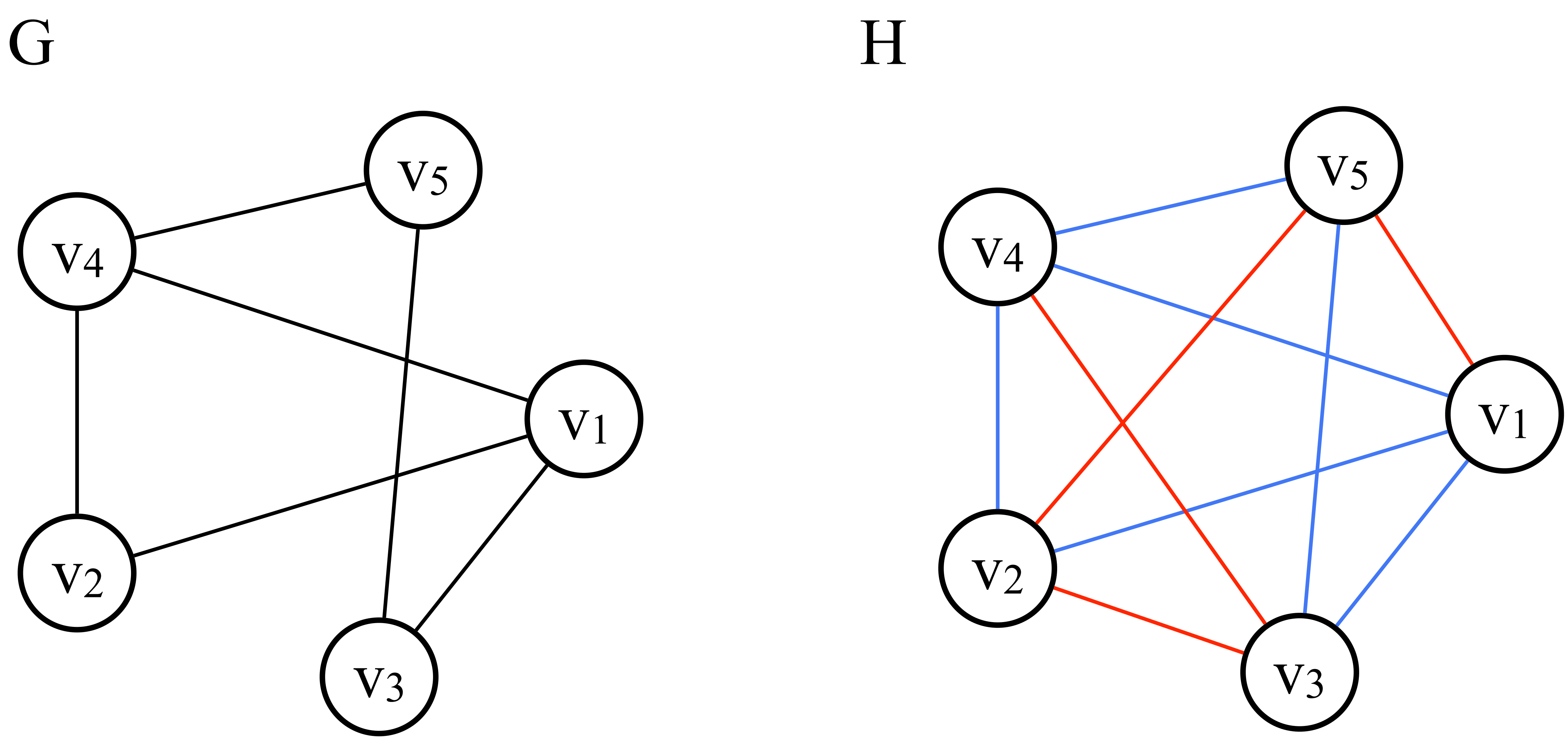
(TS-tour $\notin$ APX, the class of NP optimization problems that can be approximated to within a constant factor in polynomial time.)

Proof.

Assume for contradiction that there exists an algorithm A_ρ that can approximate TS-tour to within a constant factor ρ in polynomial time.

We claim that, for any constant $\rho \geq 1$, Hamiltonian cycle can be solved by A_ρ.

Can we approximate TS-tour without assuming that the input satisfies the triangle inequality?



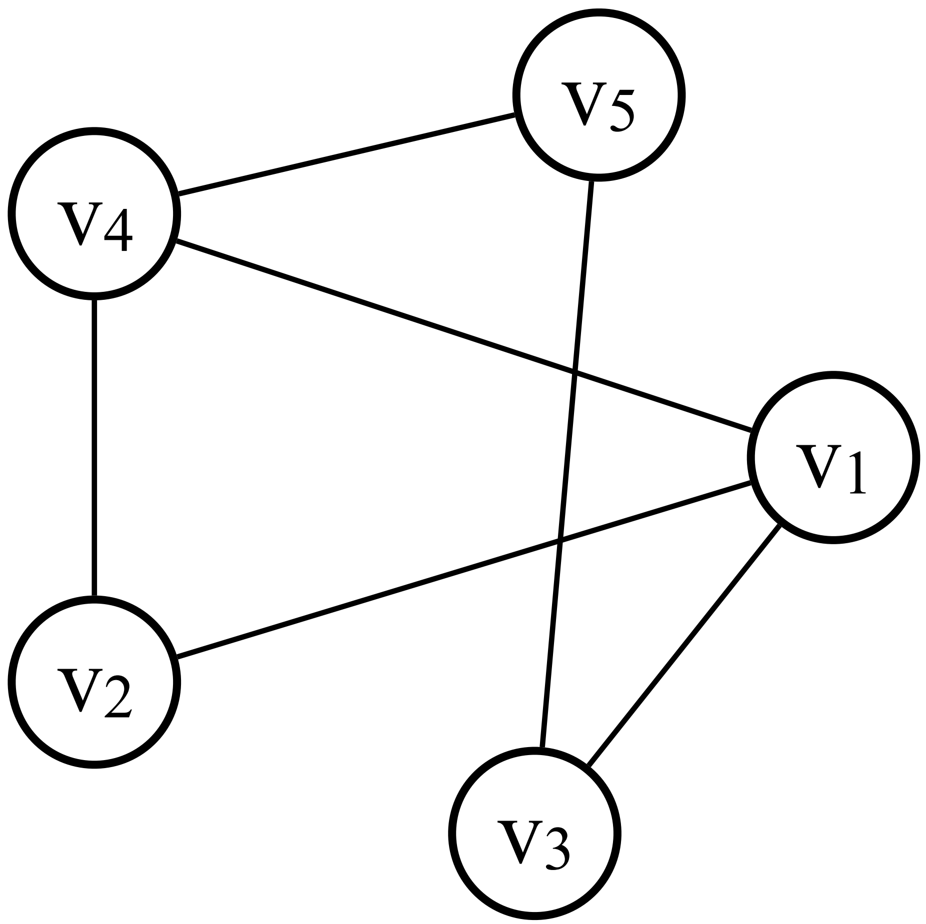
If edge (u, v) in G, add a blue edge (u, v) in H; otherwise, add a red edge (u, v) in H.

an edge with weight 1

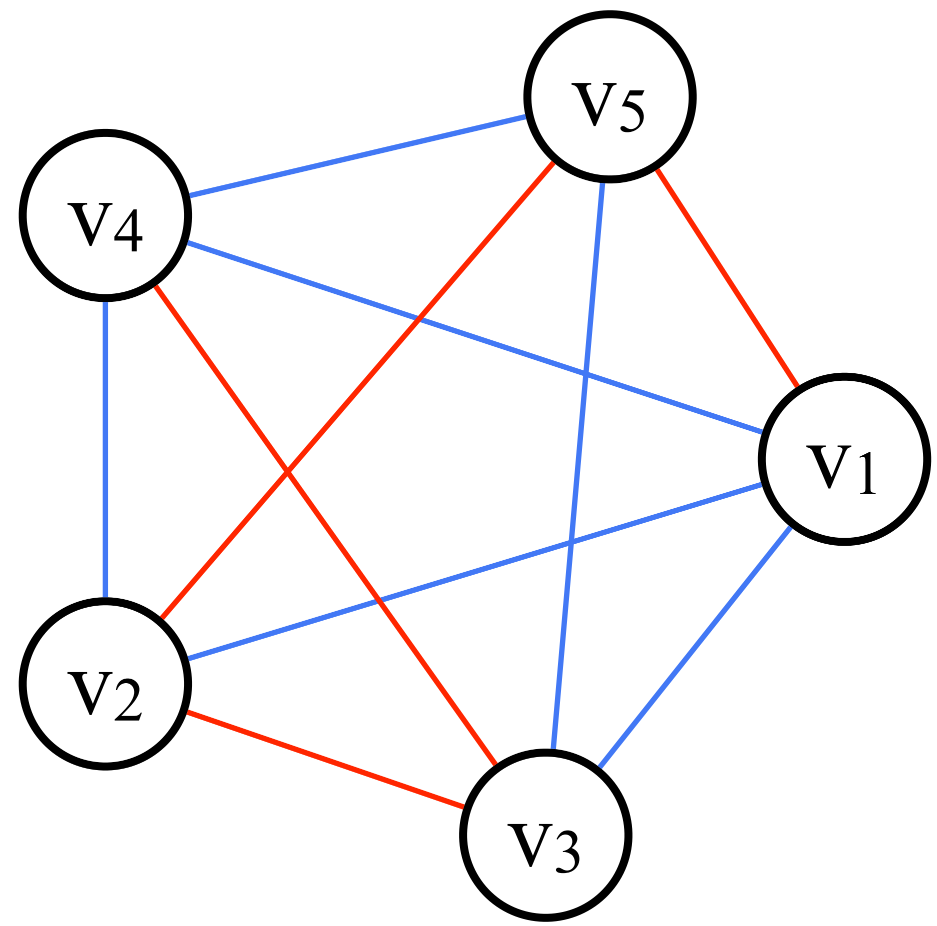
an edge with weight (ρ+1)|V|

Can we approximate TS-tour without assuming that the input satisfies the triangle inequality?

G



H



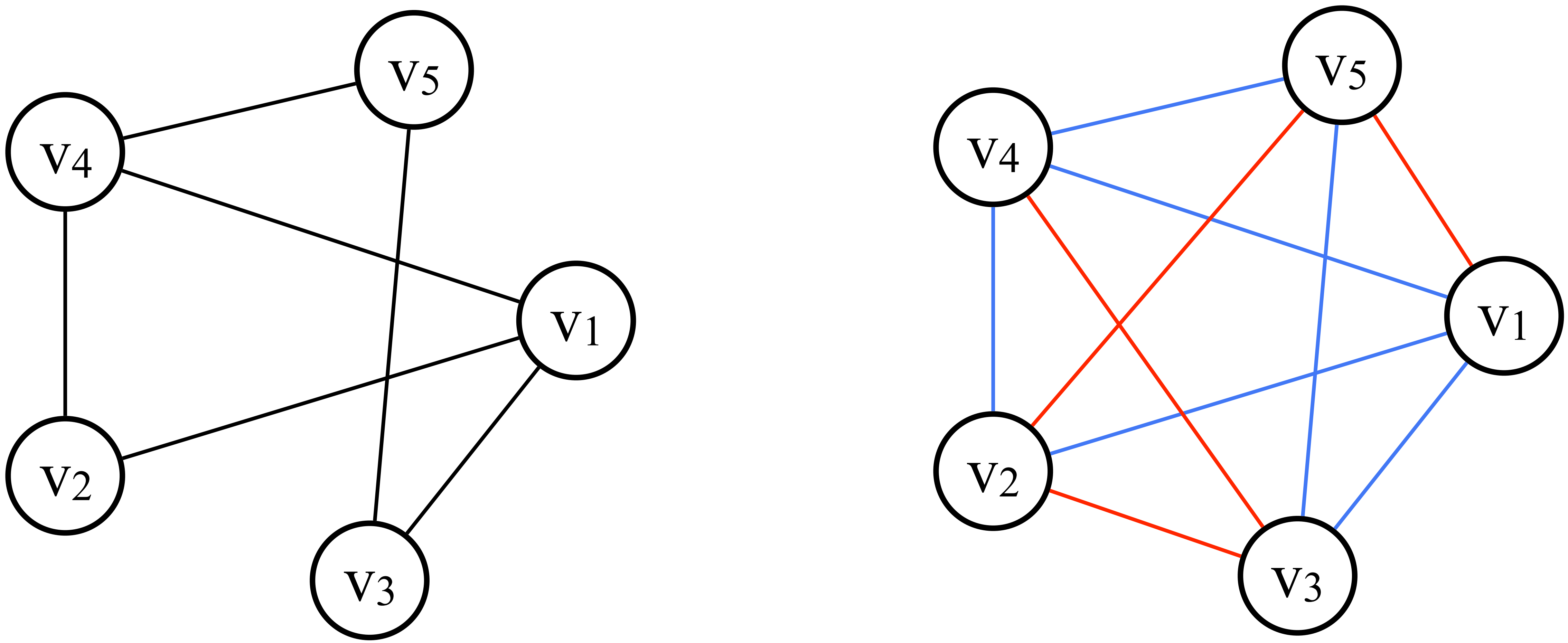
If G has an HC, then the optimal TS-tour of H has weight |V|. Algorithm A must return a value in [|V|, ρ|V|] (no red edge).

an edge with weight 1

an edge with weight (ρ+1)|V|

Can we approximate TS-tour without assuming that the input satisfies the triangle inequality?

G H



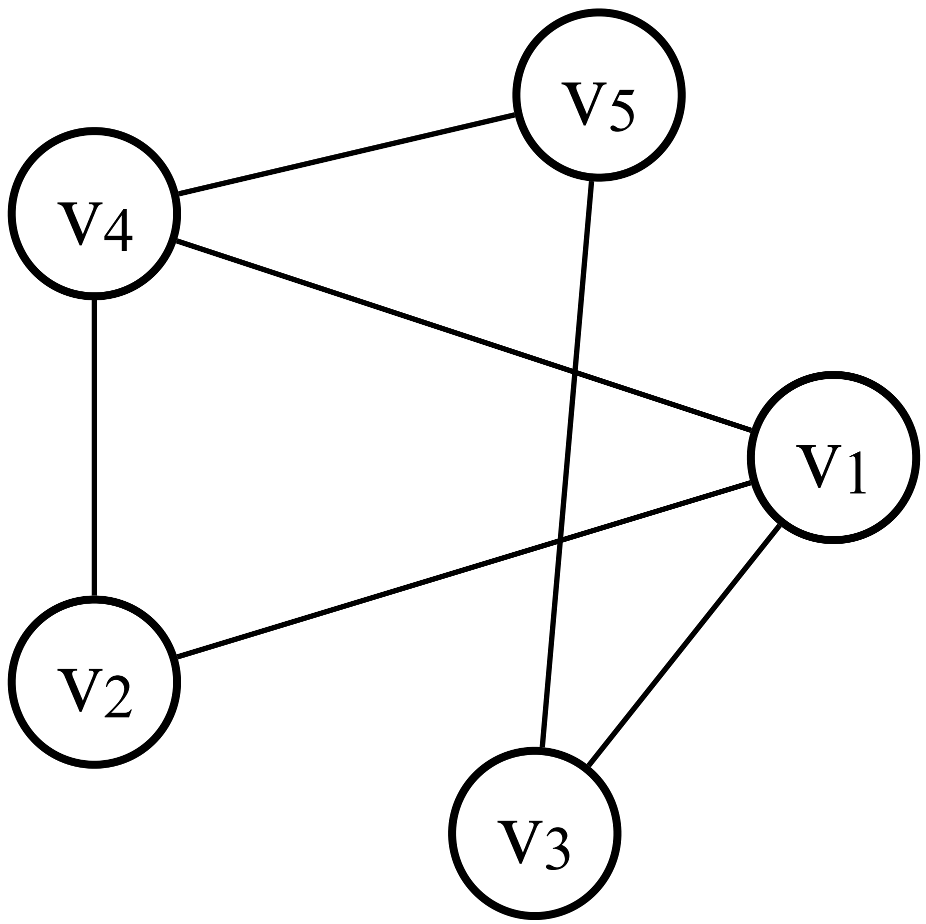
If G has no HC, then the optimal TS-tour of H has weight $\geq (\rho+1)|V|$. Algorithm A must return a value $\geq (\rho+1)|V|$.

an edge with weight 1

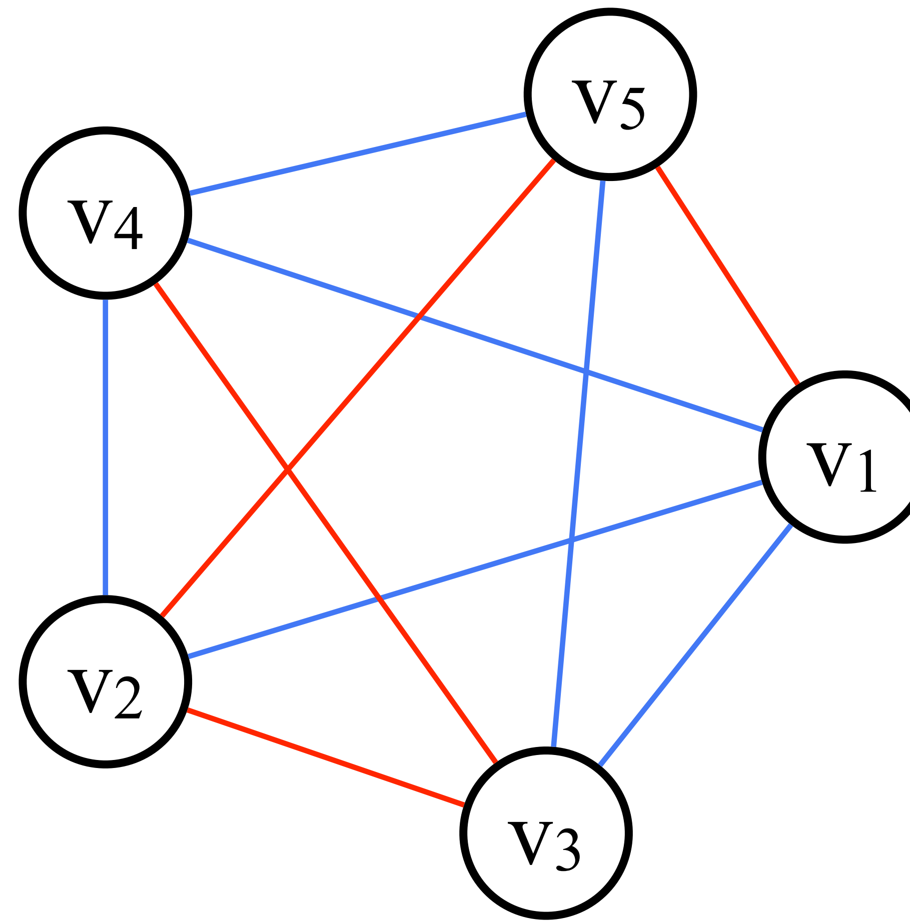
an edge with weight $(\rho+1)|V|$

Can we approximate TS-tour without assuming that the input satisfies the triangle inequality?

G



H

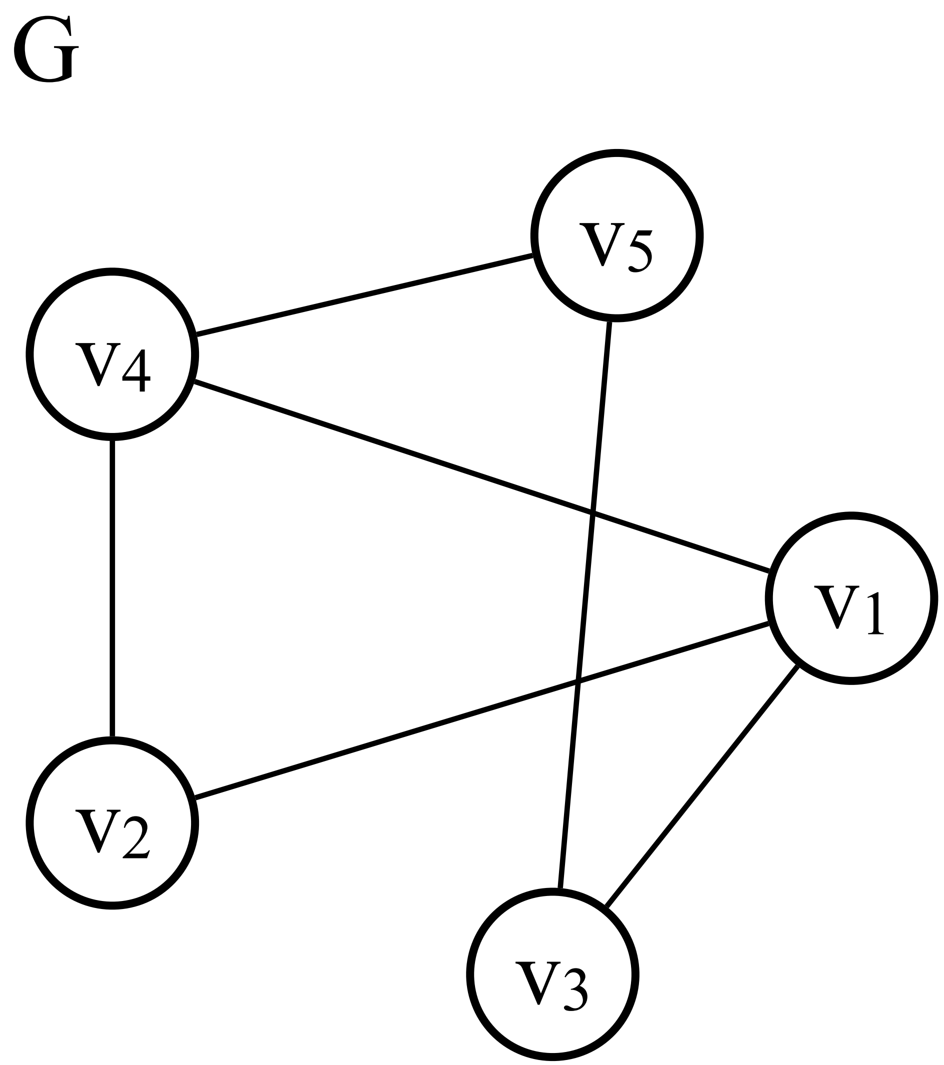


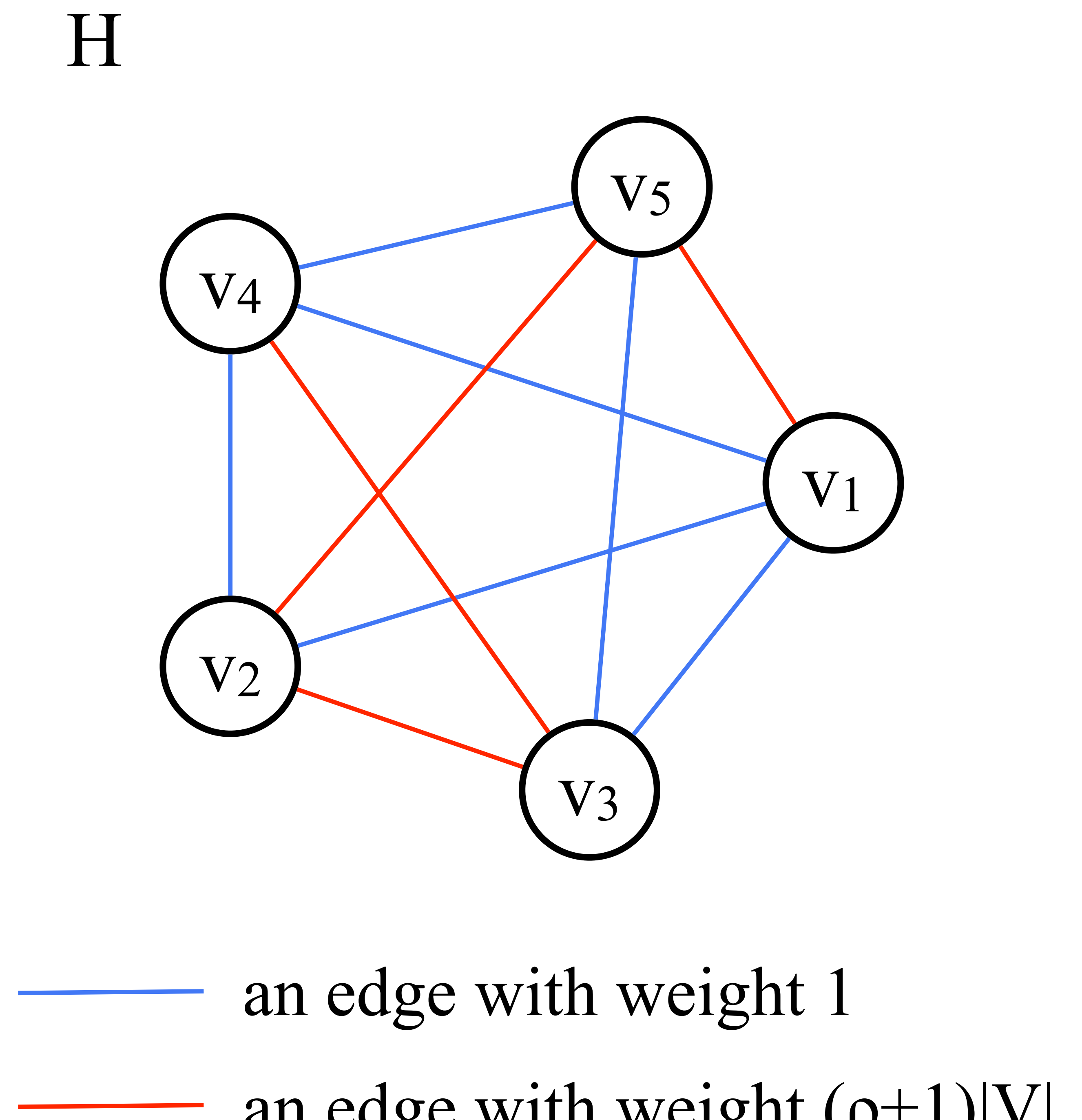
G has an HC if and only if algorithm A returns a value ≤ ρ|V|.

an edge with weight 1

an edge with weight (ρ+1)|V|

Can we approximate TS-tour without assuming that the input satisfies the triangle inequality?





TS-tour in general (may not satisfy the triangle inequality) cannot be approximated to within any constant factor unless P = NP.

Exercise

Given a directed graph G, output an acyclic subgraph H of G so that E(H) is as large as possible.

Goal: a linear-time 2-approximation algorithm.