Approximation Algorithms

Vertex Cover and TSP

Eirini Chrysikopouou

November 11, 2023

Why do we use approximation algorithms

An approximation algorithm is a way of dealing with NP-completeness for an optimization problem. This technique does not guarantee the best solution. The goal of the approximation algorithm is to come as close as possible to the optimal solution in polynomial time.

Notation

- P: Problem
- I: Instance
- $SOL_A(P, I)$: The solution we obtain for the instance I of the problem P using algorithm A.
- $OPT(P, I)$: The optimal solution for the instance I of the problem P.

Approximability

An algorithm A for a minimization problem achieves a r_A approximation factor,$(r_A : \mathbb{N} \to \mathbb{Q}+)$ if for every instance I of size $|I| = n$: $\frac{SOL_A(I)}{OPT(I)} \leq r(n)$
An algorithm A for a maximization problem Π achieves a $_A$ approximation factor, $(r_A : \mathbb{N} \to \mathbb{Q}+)$ if for every instance I of size $|I| = n$: $\frac{SOL_A(I)}{OPT(I)} \geq r(n)$

An approximation algorithm of factor ρ guarantees that the solution that the algorithm computes cannot be worse than r times the optimal solution

Minimum cardinality vertex cover

Input

A graph $G = (V, E)$

Output

A set $S \subset V$ such that S covers all edges $e \in E$ and S if of minimum size

Minimum cardinality vertex cover

Any edge algorithm

$S = \emptyset$
While E is not empty:
Pick an arbitrary edge $e = (u, v) \in E$
$S = S \cup \{u, v\}$
Remove all edges incident to either u or v from E
Return S

Approximation ratio

This is a 2-approximation algorithm. Proof:
Firstly, we have to prove that S is a vertex cover for the given graph G.Proof by contradiction.
Then we have to prove that $|S| \leq 2|OPT|$.
Let M be the set of edges selected be the algorithm. For every edge $e \in M$ both endpoints are included in S. Hence $|S| = 2|M|$.
If an edge $e = u, v \in M$ then no other edge in M has any of the of the vertices u, v as an endpoint (M is a maximal matching).
Since M is a maximal matching for G in order for all edges to be covered at least one endpoint from each edge must be included in the Vertex Cover (otherwise an edge remains uncovered). Hence,
$|M| \leq |OPT| \Rightarrow |S| = 2|M| \leq |OPT| \Rightarrow |S| \leq 2|OPT|$.

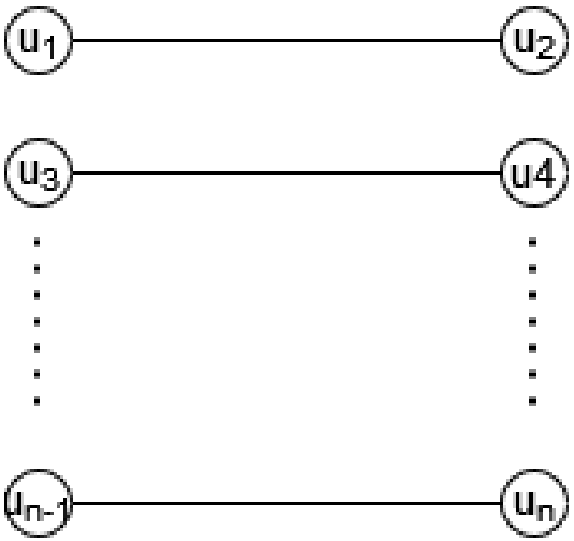
Approximation ratio

Question

Can we improve the analysis of the algorithm to get a better approximation ratio ?

To prove the analysis is tight we have to find an instance of the problem such that $\frac{|S}{||OPT|} = 2$, where S is the solution returned by our algorithm and OPT is the optimal solution for the given instance.

Approximation ratio



Our algorithm will include both endpoints for each edge$\Rightarrow |S| = n$
However $|OPT| = \frac{n}{2}$, we can include only one endpoint of each vertex to our solution and obtain a smaller Vertex Cover

Minimum Weight Vertex Cover

Input

A graph $G = (V, E)$ and a cost $w(u) \forall u \in V$

Input

Output

A set $S \subset V$ such that S covers all edges $e \in E$ and S is of minimum cost, where $cost(S) = \sum_{u \in S} w(u)$

Integer Program

The WVC problem can be expressed as an Integer Programming problem as follows:

Weighted Vertex Cover

$$\min \sum_{v \in S} w(v) x_v$$

s.t.

$$x_u + x_v \geq 1, \forall \{v, u\} \in E$$

$$x_u \in \{0, 1\}, \forall u \in V$$

But Integer Programming is $NP - Complete$. Solution: Relax the binary value restriction for x_e and transform the problem in the following LP.

LP Relaxation for WSC

Weighted Vertex Cover

$$\min \sum_{v \in S} w(v) x_v$$

s.t.

$$x_u + x_v \geq 1, \forall \{v, u\} \in E$$

$$x_u \in [0, 1], \forall u \in V$$

LP rounding

LP rounding for Vertex Cover

1 : $S = \emptyset$
2: Solve the LP for Weighted Vertex Cover.
3: Let x' be the solution
4: for i = 1 to n do
5: if $x_i' \geq \frac{1}{2}$
6: S =S $\cup \{v_i\}, (set\, x_i = 1)$
7: return S

2-approximation

Firstly, we have to prove that the solution the algorithm returns a feasible solution to the Integer program

Proof of feasibility

For any $u, v \in E$, we have $x'_v + x'_u + \geq 1$ which implies that $max(x'_v + x'_u) \geq \frac{1}{2}$
So either $x_u = 1$ or $x_v = 1$. Thus $x_u + x_v \geq 1$

This is a 2-approximation algorithm

Let x' be the solution to the LP,
$x*$ the solution returned by the algorithm
OPT the optimal solution to the problem.
$w(S) = \sum_{u \in S} w_u = \sum_{u \in V,} w(u)x*_u \leq \sum_{u \in V} w(u)2x'_u \leq 2LP - OPT \leq 2OPT$

TSP

Input

A set of $n \geq 3$ cities(vertices). To each pair of cities (i, j) we associate a weight (distance in miles, cost, etc.)

Output

Closed path (i.e., starts and ends at the same vertex) or tour of minimum cost, where the cost of the tour is defined as the sum of the weights of the edges in the tour.

Theorem

For any polynomial time computable function $a(n)$, TSP cannot be approximated within a factor of $a(n)$,unless $P = NP$.

Metric-TSP

Definition

Metric TSP is a subcase of TSP where the Triangle Inequality holds.
The Triangle Inequality:$(i,j,k) \in V(G), d(i,j) \leq d(i,k) + d(k,j)$

Claim

Consider an instance I of TSP

- Claim: $OPT(I) \geq MST(I)$
- Proof: delete one edge$e$ from an optimal solution, what remains is a spanning tree F

$OPT(I) = w(e) + C(F) \geq w(e) + MST(I) \geq MST(I)$

A simple 2-approximation algortithm

MST-based Algorithm

1. Find an MST, T, of$G$.
2. Double every edge of the MST to obtain an Eulerian graph.
3. Find an Eulerian tour, T', on this graph.
4. Output the tour that visits vertices of G in the order of their first appearance in T'. Let C be this tour

Approximation ratio

As noted above, $cost(T) \leq OPT$. Since T' contains each edge of T twice, $cost(T') = 2cost(T)$. Because of triangle inequality, after the "short- cutting" step, $cost(C) \leq cost(T)$. Combining these inequalities we get that $cost(C) \leq 2OPT$.

Christofides' Algorithm

Reminder

1. Minimum-weight matching in a weighted complete graph can be found in polynomial time.
2. A graph has a Eulerian tour if and only if all vertex degrees are even. In such a graph we can construct this tour in polynomial time.
3. The number of odd degree vertices is even in every graph G

Christofides' Algorithm

Christofides' Algorithm-1976

1.Start again by finding a minimum spanning tree, T, of cost $C(T)$
2.Find the set of vertices of T of odd degree, say S
3.Consider the graph G_S induced by S Find a minimum weight perfect matching, M, in G_S
4.Add the edges of M to T and let T be the obtained (multi)graph
5.Find an Euler cycle, W, in T
6.Find a tour H by shortcutting the Euler tour W

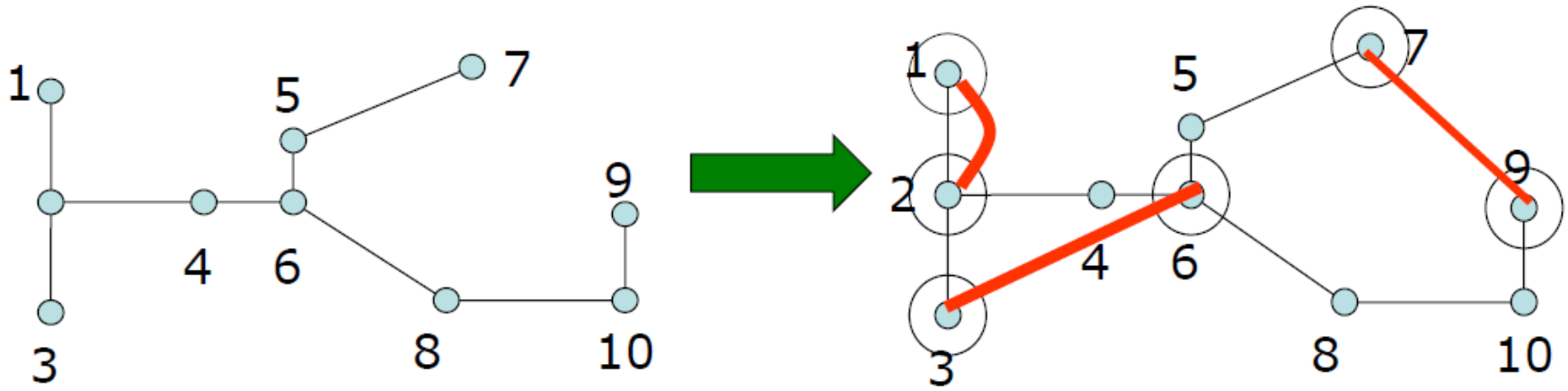
Let $V' \subset V$, such that $|V'|$ is even, and let M be a minimum cost perfect matching on V'. Then, $cost(M) \leq OPT/2$

Proof: Consider an optimal TSP tour of G, say T. Let T' be the tour on V' obtained by short-cutting T. By the triangle inequality,$cost(T') \leq cost(T)$ Now, T' is the union of two perfect matchings on V', each consisting of alternate edges of T' • Thus, the cheaper of these matchings has cost$\leq cost(T')/2 \leq OPT/2$. Hence the optimal matching also has cost at most $OPT/2$.

1.5-approximation

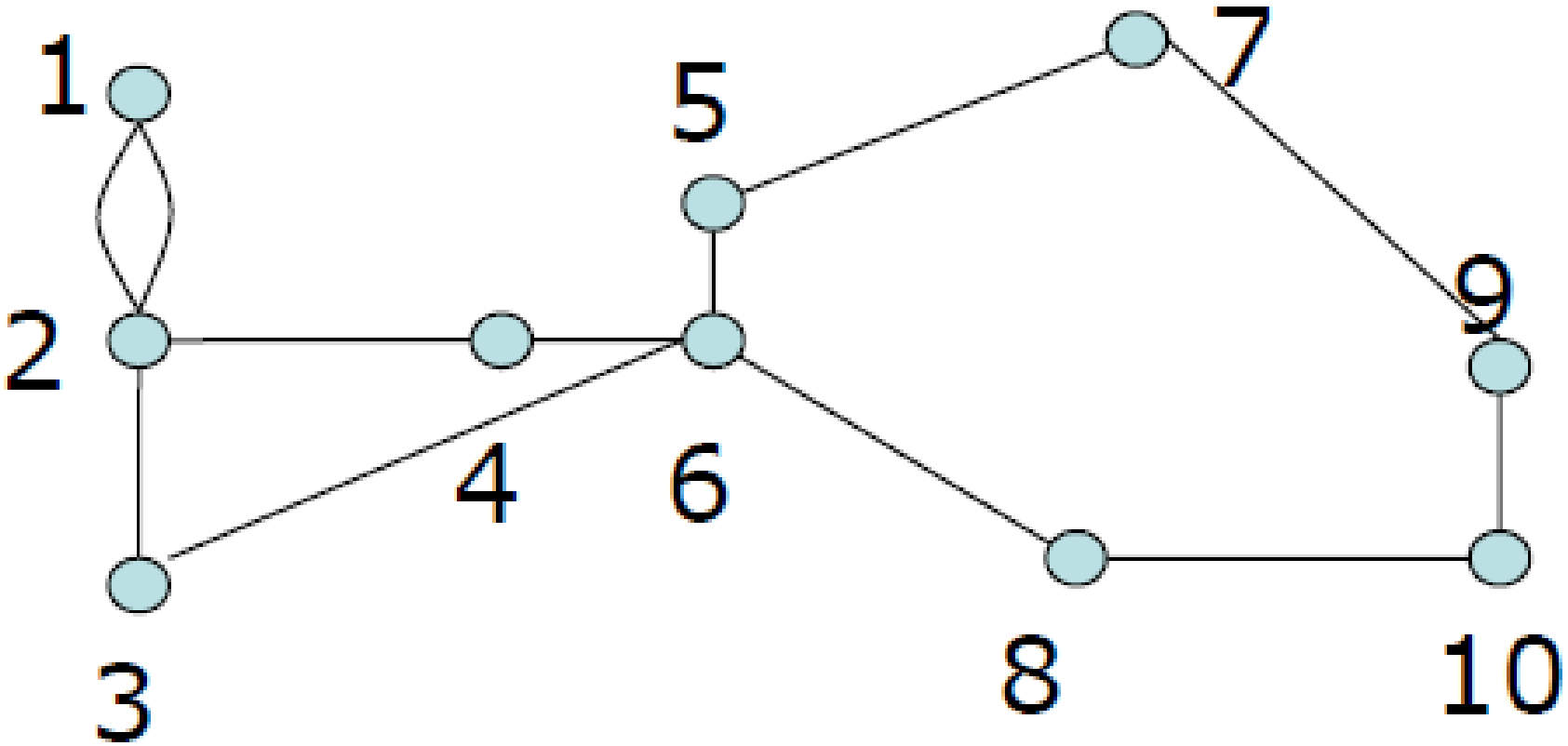
The cost of the cycle W returned by the algorithm is
$c(W) = c(T) + c(M) \leq OPT + \frac{OPT}{2} = \frac{3OPT}{2}$

Example



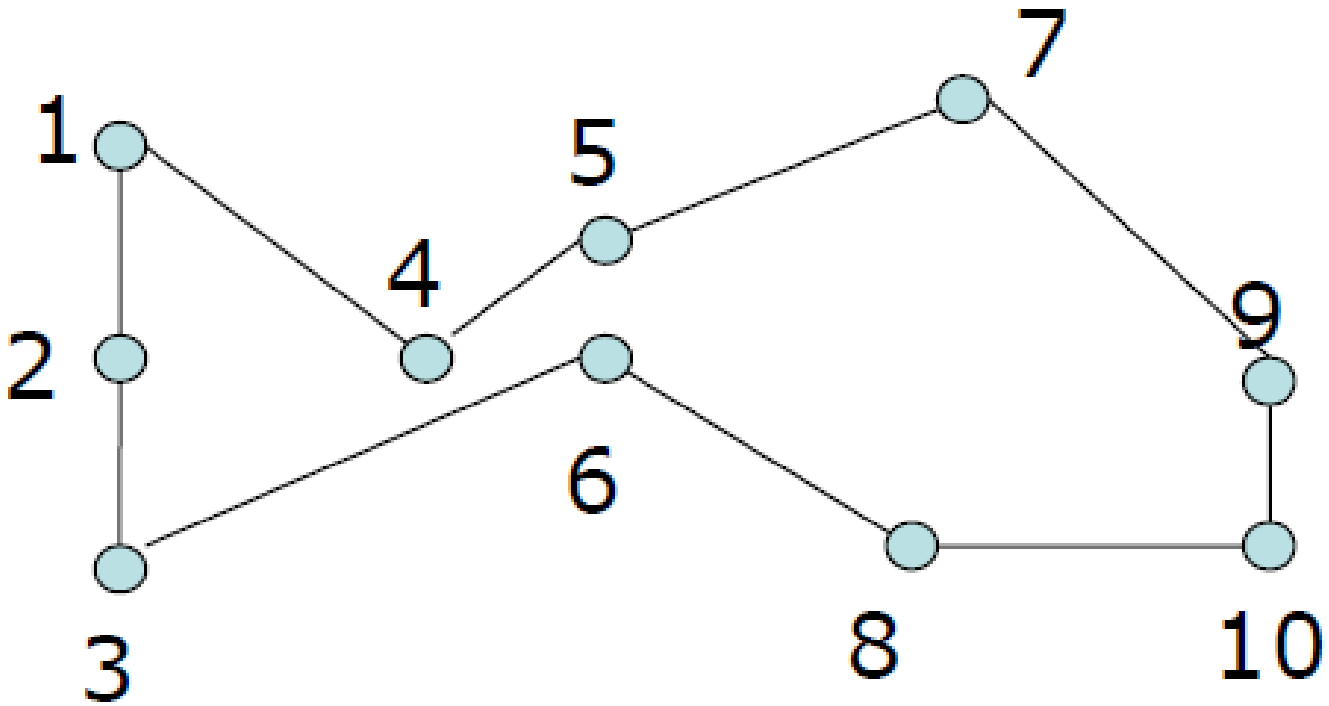
vertices with odd degree in the MST $\{1, 2, 3, 6, 7, 9\}$

Example

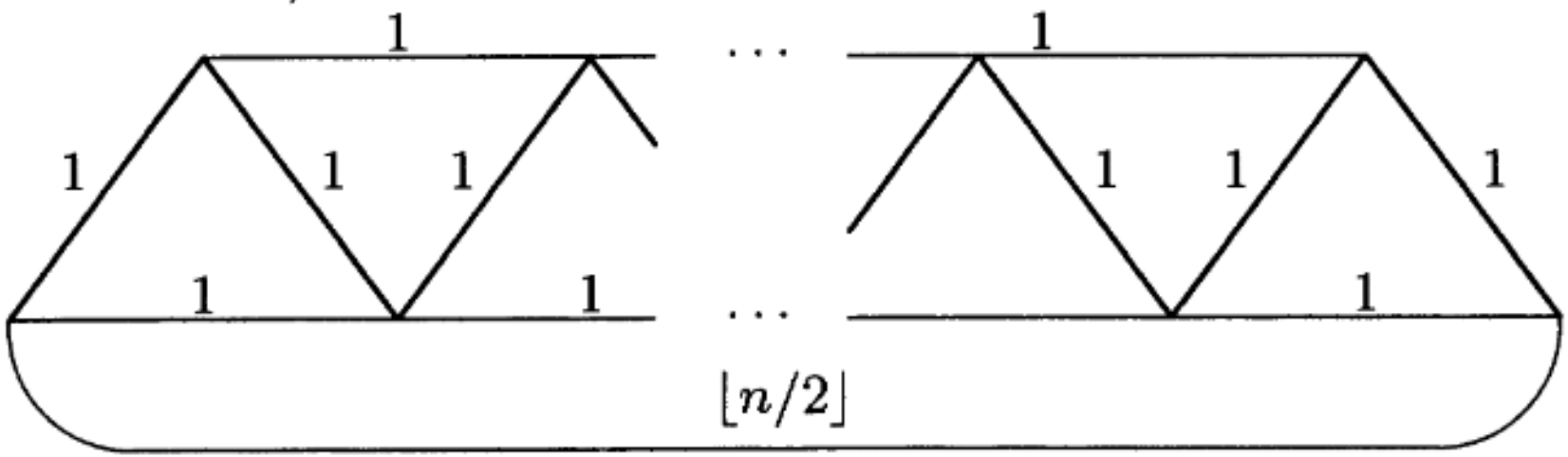


Euler cycle W : $1, 2, 3, 6, 8, 10, 9, 7, 5, 6, 4, 2, 1$

Example



H: tour obtained by shortcutting the Euler tour$W$:
H : 1, 2, 3, 6, 8, 10, 9, 7, 5, 6, 4, 2, 1



n, odd

$OPT = n$

Our algorithm may choose the zig-zag MST and the edge of weight $\frac{n}{2}$. Hence the cost C will be $(n-1) + \frac{n}{2}$

Thank you!