BILL AND NATHAN, RECORD LECTURE!!!!

BILL RECORD LECTURE!!!

TSP cannot be Approximated Unless P=NP

TSP

Notation

In this slide packet G is always a weighted graph with natural number weights

TSP

Recall TSP is the following problem

TSP

Recall TSP is the following problem

1. **Input** G and $k \in \mathbb{N}$.

TSP

Recall TSP is the following problem

1. **Input** G and $k \in \mathbb{N}$.
2. **Output** YES if there is a Ham Cycle in G of weight $\leq k$, NO otherwise.

TSP

Recall TSP is the following problem

1. **Input** G and $k \in \mathbb{N}$.
2. **Output** YES if there is a Ham Cycle in G of weight $\leq k$, NO otherwise.

This is a **Decision Problem** which has a YES-NO answer.

TSP

Recall TSP is the following problem

1. **Input** G and $k \in \mathbb{N}$.
2. **Output** YES if there is a Ham Cycle in G of weight $\leq k$, NO otherwise.

This is a **Decision Problem** which has a YES-NO answer.

What we really want is to **find** the optimal Ham Cycle.

TSP

Recall TSP is the following problem

1. **Input** G and $k \in \mathbb{N}$.
2. **Output** YES if there is a Ham Cycle in G of weight $\leq k$, NO otherwise.

This is a **Decision Problem** which has a YES-NO answer.

What we really want is to **find** the optimal Ham Cycle.

Since TSP is NPC, finding the optimal is likely hard.

TSP

Recall TSP is the following problem

1. **Input** G and $k \in \mathbb{N}$.
2. **Output** YES if there is a Ham Cycle in G of weight $\leq k$, NO otherwise.

This is a **Decision Problem** which has a YES-NO answer.

What we really want is to **find** the optimal Ham Cycle.

Since TSP is NPC, finding the optimal is likely hard.

But what about **approximating it**? Need to define this carefully.

An α-Approx For TSP

Def $\mathrm{OPT}(G)$ is the weight of the lowest weight Ham Cycle of G.

An α-Approx For TSP

Def $\mathrm{OPT}(G)$ is the weight of the lowest weight Ham Cycle of G.

Clearly if finding $\mathrm{OPT}(G)$ is in P then $\mathrm{P} = \mathrm{NP}$.

An α-Approx For TSP

Def $\mathrm{OPT}(G)$ is the weight of the lowest weight Ham Cycle of G.

Clearly if finding $\mathrm{OPT}(G)$ is in P then $\mathrm{P} = \mathrm{NP}$.

Def Let $\alpha > 1$. An **α-approx for TSP** is a poly time algorithm that, on input G, returns a cycle that is $\leq \alpha\mathrm{OPT}(G)$.

An α-Approx For TSP

Def $\mathrm{OPT}(G)$ is the weight of the lowest weight Ham Cycle of G.

Clearly if finding $\mathrm{OPT}(G)$ is in P then $\mathrm{P} = \mathrm{NP}$.

Def Let $\alpha > 1$. An **α-approx for TSP** is a poly time algorithm that, on input G, returns a cycle that is $\leq \alpha\mathrm{OPT}(G)$.

What if we can get better and better approximations?

An α-Approx For TSP

Def $\mathrm{OPT}(G)$ is the weight of the lowest weight Ham Cycle of G.

Clearly if finding $\mathrm{OPT}(G)$ is in P then $\mathrm{P} = \mathrm{NP}$.

Def Let $\alpha > 1$. An **α-approx for TSP** is a poly time algorithm that, on input G, returns a cycle that is $\le \alpha\mathrm{OPT}(G)$.

What if we can get better and better approximations?

Def A **Poly time Approx Scheme (PTAS)** for TSP is a poly time algorithm that, on input (G, ϵ), returns a cycle that is $\le (1+\epsilon)\mathrm{OPT}(G)$. Run time depends on ϵ. Can be bad: $n^{2^{1/\epsilon^2}}$.

An α-Approx For TSP

Def $\mathrm{OPT}(G)$ is the weight of the lowest weight Ham Cycle of G.

Clearly if finding $\mathrm{OPT}(G)$ is in P then $\mathrm{P} = \mathrm{NP}$.

Def Let $\alpha > 1$. An **α-approx for TSP** is a poly time algorithm that, on input G, returns a cycle that is $\le \alpha\mathrm{OPT}(G)$.

What if we can get better and better approximations?

Def A **Poly time Approx Scheme (PTAS)** for TSP is a poly time algorithm that, on input (G, ϵ), returns a cycle that is $\le (1+\epsilon)\mathrm{OPT}(G)$. Run time depends on ϵ. Can be bad: $n^{2^{1/\epsilon^2}}$.

VOTE assuming $\mathrm{P} \ne \mathrm{NP}$.

An α-Approx For TSP

Def $\mathrm{OPT}(G)$ is the weight of the lowest weight Ham Cycle of G.

Clearly if finding $\mathrm{OPT}(G)$ is in P then $\mathrm{P} = \mathrm{NP}$.

Def Let $\alpha > 1$. An **α-approx for TSP** is a poly time algorithm that, on input G, returns a cycle that is $\le \alpha \mathrm{OPT}(G)$.

What if we can get better and better approximations?

Def A **Poly time Approx Scheme (PTAS)** for TSP is a poly time algorithm that, on input (G, ϵ), returns a cycle that is $\le (1+\epsilon)\mathrm{OPT}(G)$. Run time depends on ϵ. Can be bad: $n^{2^{1/\epsilon^2}}$.

VOTE assuming $\mathrm{P} \ne \mathrm{NP}$.

1) There is a PTAS for TSP.

An α-Approx For TSP

Def $\mathrm{OPT}(G)$ is the weight of the lowest weight Ham Cycle of G.

Clearly if finding $\mathrm{OPT}(G)$ is in P then $\mathrm{P} = \mathrm{NP}$.

Def Let $\alpha > 1$. An **α-approx for TSP** is a poly time algorithm that, on input G, returns a cycle that is $\le \alpha \mathrm{OPT}(G)$.

What if we can get better and better approximations?

Def A **Poly time Approx Scheme (PTAS)** for TSP is a poly time algorithm that, on input (G, ϵ), returns a cycle that is $\le (1+\epsilon)\mathrm{OPT}(G)$. Run time depends on ϵ. Can be bad: $n^{2^{1/\epsilon^2}}$.

VOTE assuming $\mathrm{P} \ne \mathrm{NP}$.

1) There is a PTAS for TSP.

2) There is an α such that (1) TSP has an α-approx but (2) for all $\beta < \alpha$ there is no β-approx for TSP.

An α-Approx For TSP

Def $\mathrm{OPT}(G)$ is the weight of the lowest weight Ham Cycle of G.

Clearly if finding $\mathrm{OPT}(G)$ is in P then $\mathrm{P} = \mathrm{NP}$.

Def Let $\alpha > 1$. An **α-approx for TSP** is a poly time algorithm that, on input G, returns a cycle that is $\le \alpha \mathrm{OPT}(G)$.

What if we can get better and better approximations?

Def A **Poly time Approx Scheme (PTAS)** for TSP is a poly time algorithm that, on input (G, ϵ), returns a cycle that is $\le (1+\epsilon)\mathrm{OPT}(G)$. Run time depends on ϵ. Can be bad: $n^{2^{1/\epsilon^2}}$.

VOTE assuming $\mathrm{P} \ne \mathrm{NP}$.

1) There is a PTAS for TSP.

2) There is an α such that (1) TSP has an α-approx but (2) for all $\beta < \alpha$ there is no β-approx for TSP.

3) There is no such α. E.g., there is no $(1 + \frac{1}{2^{1000}})$-approx for TSP.

An α-Approx For TSP

Def $\mathrm{OPT}(G)$ is the weight of the lowest weight Ham Cycle of G.

Clearly if finding $\mathrm{OPT}(G)$ is in P then $\mathrm{P} = \mathrm{NP}$.

Def Let $\alpha > 1$. An **α-approx for TSP** is a poly time algorithm that, on input G, returns a cycle that is $\le \alpha \mathrm{OPT}(G)$.

What if we can get better and better approximations?

Def A **Poly time Approx Scheme (PTAS)** for TSP is a poly time algorithm that, on input (G, ϵ), returns a cycle that is $\le (1+\epsilon)\mathrm{OPT}(G)$. Run time depends on ϵ. Can be bad: $n^{2^{1/\epsilon^2}}$.

VOTE assuming $\mathrm{P} \ne \mathrm{NP}$.

1) There is a PTAS for TSP.

2) There is an α such that (1) TSP has an α-approx but (2) for all $\beta < \alpha$ there is no β-approx for TSP.

3) There is no such α. E.g., there is no $(1 + \frac{1}{2^{1000}})$-approx for TSP.

ANSWER: 3, no approx. But there is approx for subcases.

Approximating TSP

Approximating TSP

1. **Metric TSP**: TSP problem restricted to weighted graphs that are symmetric and satisfy the triangle inequality: $w(x, y) + w(y, z) \geq w(x, z)$. Christofides (1976) and Serdyukov (1978) gives a $\frac{3}{2}$-approximation to metric TSP.

Approximating TSP

1. **Metric TSP**: TSP problem restricted to weighted graphs that are symmetric and satisfy the triangle inequality: $w(x, y) + w(y, z) \geq w(x, z)$. Christofides (1976) and Serdyukov (1978) gives a $\mathbf{\frac{3}{2}}$-approximation to metric TSP.
2. Karlan, Klein, Oveis-Gharan (2020) got the first improvement over $\frac{3}{2}$-approx: a $(\mathbf{\frac{3}{2}} - \boldsymbol{\epsilon})$-approx to the metric TSP ($\epsilon < 10^{-36}$).

Approximating TSP

1. **Metric TSP**: TSP problem restricted to weighted graphs that are symmetric and satisfy the triangle inequality: $w(x, y) + w(y, z) \geq w(x, z)$. Christofides (1976) and Serdyukov (1978) gives a $\frac{3}{2}$-approximation to metric TSP.
2. Karlan, Klein, Oveis-Gharan (2020) got the first improvement over $\frac{3}{2}$-approx: a $(\frac{3}{2} - \epsilon)$-approx to the metric TSP $(\epsilon < 10^{-36})$.
3. **Euclidean TSP**: TSP problem when the graph is a set of points in the plane and the weights are the Euclidean distances.

Approximating TSP

1. **Metric TSP**: TSP problem restricted to weighted graphs that are symmetric and satisfy the triangle inequality: $w(x, y) + w(y, z) \geq w(x, z)$. Christofides (1976) and Serdyukov (1978) gives a $\frac{3}{2}$-approximation to metric TSP.
2. Karlan, Klein, Oveis-Gharan (2020) got the first improvement over $\frac{3}{2}$-approx: a $(\frac{3}{2} - \epsilon)$-approx to the metric TSP ($\epsilon < 10^{-36}$).
3. **Euclidean TSP**: TSP problem when the graph is a set of points in the plane and the weights are the Euclidean distances. Arora (1998) and Mitchell (1999) showed that, for all ϵ, there is an $(1 + \epsilon)$-approximation in time $O(n(\log n)^{O(1/\epsilon)})$.

Approximating TSP

1. **Metric TSP**: TSP problem restricted to weighted graphs that are symmetric and satisfy the triangle inequality: $w(x,y) + w(y,z) \geq w(x,z)$. Christofides (1976) and Serdyukov (1978) gives a $\frac{3}{2}$-approximation to metric TSP.
2. Karlan, Klein, Oveis-Gharan (2020) got the first improvement over $\frac{3}{2}$-approx: a $(\frac{3}{2} - \epsilon)$-approx to the metric TSP ($\epsilon < 10^{-36}$).
3. **Euclidean TSP**: TSP problem when the graph is a set of points in the plane and the weights are the Euclidean distances. Arora (1998) and Mitchell (1999) showed that, for all ϵ, there is an $(1+\epsilon)$-approximation in time $O(n(\log n)^{O(1/\epsilon)})$.
4. Arora and Mitchell actually have an algorithm that works on n points in R^d that runs in time $O(n(\log n)^{O(\sqrt{d}/\epsilon)^{d-1}})$.

TSP Does Not have an α-Approx

If TSP has an approx then HAMC is in P

Assume TSP has an α-approx via Program M. $\alpha > 1$.

If TSP has an approx then HAMC is in P

Assume TSP has an α-approx via Program M. $\alpha > 1$.
1) Input G, an unweighted Graph on n vertices.

If TSP has an approx then HAMC is in P

Assume TSP has an α-approx via Program M. $\alpha > 1$.
1) Input G, an unweighted Graph on n vertices.
2) Let G' be the weighed graph where every edge in G has weight 1 and every non-edge has weight B where we determine B later.

If TSP has an approx then HAMC is in P

Assume TSP has an α-approx via Program M. $\alpha > 1$.
1) Input G, an unweighted Graph on n vertices.
2) Let G' be the weighed graph where every edge in G has weight 1 and every non-edge has weight B where we determine B later.

Comment

If G has a HAMC then $\mathrm{OPT}(G') \leq n$.

If TSP has an approx then HAMC is in P

Assume TSP has an α-approx via Program M. $\alpha > 1$.
1) Input G, an unweighted Graph on n vertices.
2) Let G' be the weighed graph where every edge in G has weight 1 and every non-edge has weight B where we determine B later.

Comment

If G has a HAMC then $\mathrm{OPT}(G') \le n$.
If G has no HAMC then $\mathrm{OPT}(G') \ge B$.

If TSP has an approx then HAMC is in P

Assume TSP has an α-approx via Program M. $\alpha > 1$.
1) Input G, an unweighted Graph on n vertices.
2) Let G' be the weighed graph where every edge in G has weight 1 and every non-edge has weight B where we determine B later.
Comment
If G has a HAMC then $\mathrm{OPT}(G') \le n$.
If G has no HAMC then $\mathrm{OPT}(G') \ge B$.
3) Run the α-approx on G'.

If TSP has an approx then HAMC is in P

Assume TSP has an α-approx via Program M. $\alpha > 1$.
1) Input G, an unweighted Graph on n vertices.
2) Let G' be the weighed graph where every edge in G has weight 1 and every non-edge has weight B where we determine B later.

Comment

If G has a HAMC then $\mathrm{OPT}(G') \le n$.
If G has no HAMC then $\mathrm{OPT}(G') \ge B$.
3) Run the α-approx on G'.

Comment

If G has a HAMC then $\mathrm{OPT}(G') \le n$ so $M(G') \le \alpha n$.

If TSP has an approx then HAMC is in P

Assume TSP has an α-approx via Program M. $\alpha > 1$.
1) Input G, an unweighted Graph on n vertices.
2) Let G' be the weighed graph where every edge in G has weight 1 and every non-edge has weight B where we determine B later.
Comment
If G has a HAMC then $\mathrm{OPT}(G') \le n$.
If G has no HAMC then $\mathrm{OPT}(G') \ge B$.
3) Run the α-approx on G'.
Comment
If G has a HAMC then $\mathrm{OPT}(G') \le n$ so $M(G') \le \alpha n$.
If G has no HAMC then $\mathrm{OPT}(G') \ge B$ so $M(G') \ge B$.

If TSP has an approx then HAMC is in P

Assume TSP has an α-approx via Program M. $\alpha > 1$.
1) Input G, an unweighted Graph on n vertices.
2) Let G' be the weighed graph where every edge in G has weight 1 and every non-edge has weight B where we determine B later.

Comment

If G has a HAMC then $\mathrm{OPT}(G') \le n$.
If G has no HAMC then $\mathrm{OPT}(G') \ge B$.
3) Run the α-approx on G'.

Comment

If G has a HAMC then $\mathrm{OPT}(G') \le n$ so $M(G') \le \alpha n$.
If G has no HAMC then $\mathrm{OPT}(G') \ge B$ so $M(G') \ge B$.
Need to set B such that $\alpha n < B$. $B = n^2$ will suffice.

If TSP has an approx then HAMC is in P

Assume TSP has an α-approx via Program M. $\alpha > 1$.
1) Input G, an unweighted Graph on n vertices.
2) Let G' be the weighed graph where every edge in G has weight 1 and every non-edge has weight B where we determine B later.

Comment

If G has a HAMC then $\mathrm{OPT}(G') \le n$.
If G has no HAMC then $\mathrm{OPT}(G') \ge B$.
3) Run the α-approx on G'.

Comment

If G has a HAMC then $\mathrm{OPT}(G') \le n$ so $M(G') \le \alpha n$.
If G has no HAMC then $\mathrm{OPT}(G') \ge B$ so $M(G') \ge B$.
Need to set B such that $\alpha n < B$. $B = n^2$ will suffice.
4)
Case 1: If $M(G') \le \alpha n$ then output YES.

If TSP has an approx then HAMC is in P

Assume TSP has an α-approx via Program M. $\alpha > 1$.
1) Input G, an unweighted Graph on n vertices.
2) Let G' be the weighed graph where every edge in G has weight 1 and every non-edge has weight B where we determine B later.

Comment

If G has a HAMC then $\mathrm{OPT}(G') \le n$.
If G has no HAMC then $\mathrm{OPT}(G') \ge B$.
3) Run the α-approx on G'.

Comment

If G has a HAMC then $\mathrm{OPT}(G') \le n$ so $M(G') \le \alpha n$.
If G has no HAMC then $\mathrm{OPT}(G') \ge B$ so $M(G') \ge B$.
Need to set B such that $\alpha n < B$. $B = n^2$ will suffice.
4)
Case 1: If $M(G') \le \alpha n$ then output YES.
Case 2: If $M(G') \ge B$ then output NO.

We can Do Better

We showed:

Thm Let $\alpha \geq 1$. If there is an α-approx for TSP then P=NP.

We can Do Better

We showed:
Thm Let $\alpha \geq 1$. If there is an α-approx for TSP then P=NP.

If you look at the proof more carefully you can prove this:
Thm Let $\alpha(n)$ be a polynomial. If there is an $\alpha(n)$-approx for TSP then P=NP.

Summary of Other Non-Approx Results

History: 1971-1997

History: 1971-1997

1. The TSP result goes back before 1978 and is folklore.

History: 1971-1997

1. The TSP result goes back before 1978 and is folklore.
2. Before 1990 there were a few other non-approx results: Ind set, Coloring, Knapsack, Prob others. All had elementary though clever proofs, like the TSP result.

History: 1971-1997

1. The TSP result goes back before 1978 and is folklore.
2. Before 1990 there were a few other non-approx results: Ind set, Coloring, Knapsack, Prob others. All had elementary though clever proofs, like the TSP result.
3. In 1991 a paper came out that showed:

History: 1971-1997

1. The TSP result goes back before 1978 and is folklore.
2. Before 1990 there were a few other non-approx results: Ind set, Coloring, Knapsack, Prob others. All had elementary though clever proofs, like the TSP result.
3. In 1991 a paper came out that showed:
   3.1 Many results like: **f has a PTAS IFF g has a PTAS**.

History: 1971-1997

1. The TSP result goes back before 1978 and is folklore.
2. Before 1990 there were a few other non-approx results: Ind set, Coloring, Knapsack, Prob others. All had elementary though clever proofs, like the TSP result.
3. In 1991 a paper came out that showed:
   3.1 Many results like: ***f* has a PTAS IFF *g* has a PTAS**.
   3.2 A class MAXSNP of functions that seemed to not have PTAS was defined.

History: 1971-1997

1. The TSP result goes back before 1978 and is folklore.
2. Before 1990 there were a few other non-approx results: Ind set, Coloring, Knapsack, Prob others. All had elementary though clever proofs, like the TSP result.
3. In 1991 a paper came out that showed:
   3.1 Many results like: ***f* has a PTAS IFF *g* has a PTAS**.
   3.2 A class MAXSNP of functions that seemed to not have PTAS was defined.
   3.3 The problem:
   $\mathrm{MAX3SAT}(\phi) =$ max numb of clauses that can be satisfied was shown complete for MAXSNP.

History: 1971-1997

1. The TSP result goes back before 1978 and is folklore.
2. Before 1990 there were a few other non-approx results: Ind set, Coloring, Knapsack, Prob others. All had elementary though clever proofs, like the TSP result.
3. In 1991 a paper came out that showed:
   3.1 Many results like: ***f* has a PTAS IFF *g* has a PTAS**.
   3.2 A class MAXSNP of functions that seemed to not have PTAS was defined.
   3.3 The problem:
   $\mathrm{MAX3SAT}(\phi) =$ max numb of clauses that can be satisfied was shown complete for MAXSNP.
   But this was not very satisfying: it is plausible all these problems in MAXSNP had a PTAS.

History: 1998-2021

History: 1998-2021

1. Motivated by (among other things) trying to find lower bounds on approx, the class $\mathrm{PCP}(q(n), r(n)\epsilon(n))$ was defined.

History: 1998-2021

1. Motivated by (among other things) trying to find lower bounds on approx, the class $\mathrm{PCP}(q(n), r(n)\epsilon(n))$ was defined.
2. In 1998 it was shown that $\mathrm{NP} = \mathrm{PCP}(O(1), O(\log n), \frac{1}{n})$. This implied (with a lot of additional work):

History: 1998-2021

1. Motivated by (among other things) trying to find lower bounds on approx, the class $\mathrm{PCP}(q(n), r(n)\epsilon(n))$ was defined.
2. In 1998 it was shown that $\mathrm{NP} = \mathrm{PCP}(O(1), O(\log n), \frac{1}{n})$. This implied (with a lot of additional work):
   2.1 If MAX3SAT has a PTAS then $\mathrm{P} = \mathrm{NP}$.

History: 1998-2021

1. Motivated by (among other things) trying to find lower bounds on approx, the class $\mathrm{PCP}(q(n), r(n)\epsilon(n))$ was defined.
2. In 1998 it was shown that $\mathrm{NP} = \mathrm{PCP}(O(1), O(\log n), \frac{1}{n})$. This implied (with a lot of additional work):
   2.1 If MAX3SAT has a PTAS then $\mathrm{P} = \mathrm{NP}$.
   2.2 If CLIQ can be **well approximated** then $\mathrm{P} = \mathrm{NP}$.

History: 1998-2021

1. Motivated by (among other things) trying to find lower bounds on approx, the class $\mathrm{PCP}(q(n), r(n)\epsilon(n))$ was defined.
2. In 1998 it was shown that $\mathrm{NP} = \mathrm{PCP}(O(1), O(\log n), \frac{1}{n})$. This implied (with a lot of additional work):
   2.1 If MAX3SAT has a PTAS then $\mathrm{P} = \mathrm{NP}$.
   2.2 If CLIQ can be **well approximated** then $\mathrm{P} = \mathrm{NP}$.
   2.3 If SET COVER has an $(1 - o(1))\ln(n)$ approx then $\mathrm{P} = \mathrm{NP}$. (It is known to have a $\ln(n)$-approx. This took about 10 papers with many intermediary results.