QSAT: Given φ, determine whether $\exists x_1 \forall x_2 \exists x_3 \dots \varphi(x_1, \dots, x_n)$ is true.

$x_1 = F$

$x_2 = T$ $x_2 = F$

$x_3 = F$ $x_3 = T$

leaves $\approx 2^{\frac{n}{2}}$ (2 branches every other level)

$(x_1 \wedge x_2 \wedge x_3) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge \neg x_2 \wedge x_3)$

T F T/F F F

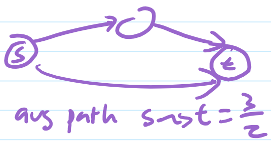
F T F T T T T

T T F

you lose!

QSAT $\in$ NP?? not by this verification alg

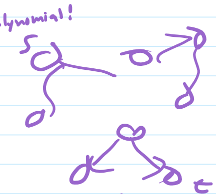
AVG-PATH-LEN: Given directed G, vertices s, t, integer k, determine if avg path length $s \rightsquigarrow t \leq k$



avg path $s \rightsquigarrow t = \frac{3}{2}$

AVG-PATH-LEN $\in$ NP??? not by simple verification alg

evidence = list of paths + their lengths — not polynomial!

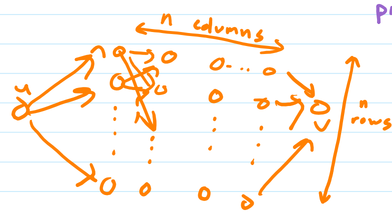


how to verify claim that avg $s \rightsquigarrow t$ path length ≤ 10?

AVG-PATH-LEN is NP-hard (for all $X \in NP$, $X \leq_p$ A-P-L)

st-DIR-HAM-PATH $\leq_p$ AVG-PATH-LEN

given G, replace every $u \longrightarrow v$ with



one $s \rightsquigarrow t$ HP in G becomes $(n^n)^{n-1}$ paths of length $(n-1)(n+1)$

all other paths in G become $\leq (n^n)^{n-2}$ paths of length $\leq (n-2)(n+1)$

new graph has avg. path len. $> (n-2)(n+1) \longleftrightarrow$ G has a HP (b/c so many long paths from a single HP in G)

$n = 10^{10}$ one HP $\rightarrow (10^{10})^{9}$ paths of len 99

$\leq 10 \cdot 10!$ $(10^{10})^{8}$ of len ≤ 88, ≥ 11

$10^{10} \gg 10 \cdot 10!$, so avg very close to 99

PSPACE: set of problems that can be solved in poly space

QSAT $\in$ PSPACE eval(φ, Q, i)

$Q' \leftarrow$ other quantifier

$\varphi_T \leftarrow \varphi$ with x_i set to T

$\varphi_F \leftarrow \varphi$ " " F

$r_T \leftarrow$ eval(φ_T, Q', i+1)

$r_F \leftarrow$ eval(φ_F, Q', i+1)

if $Q = \forall$ and r_T and r_F both T
 return T
else if $Q = \exists$ and r_T or r_F T
 return T
else
 return F

linear storage per level

linear # levels

total space quadratic (polynomial)

AVG-PATH-LEN $\in$ PSPACE

go through all paths
add len of each to running total, incr count
return $\frac{\text{tot}}{\text{count}} \leq k$

P $\subseteq$ PSPACE

bound on time = bound on space b/c O(1) storage added per step

NP ⊆ PSPACE
co-NP ⊆ PSPACE

Let $X \in NP$ [want: $X \in PSPACE$]
Then $\exists$ poly-time verifier X-VERIFY

X(s)
O(1) verified = F
poly for each y
verified ← verified | X-VERIFY(x, y) poly-time, so poly-space
return verified

P $\overset{?}{=}$ PSPACE
NP $\overset{?}{=}$ PSPACE

open questions

QSAT is PSPACE-complete (for all $X \in PSPACE$, $X \leq_p QSAT$)

Approximation Algorithms

FIND-VERTEX-COVER : given undirected G, output smallest vertex cover

a 2-approximation algorithm always outputs a VC of size $\leq 2 \cdot |C^*|$

smallest vertex cover

APPROX-FIND-VERTEX-COVER(G)

$C \leftarrow \emptyset$ the cover
$A \leftarrow \emptyset$
$E' \leftarrow E$ uncovered edges
while $E' \neq \emptyset$
pick (u,v) from E'
preserves c) ⟶ $C \leftarrow C \cup \{u,v\}$
$A \leftarrow A \cup \{(u,v)\}$] preserves a)
preserves b)
remove from E' all edges w/ u as endpoint
and all w/ v as endpoint
return C

INVARIANT: a) $|C| = 2 \cdot |A|$

b) edges in A have no endpoints in common w/ each other or edges in E'

c) C covers edges not in E'

$|\overbrace{C^*}^{\text{optimal cover}}| \geq |A|$ C^* contains ≥ 1 endpoint of each $(u,v) \in A$
since endpoints in A are unique

$|C| = 2|A| \leq 2 \cdot |C^*|$

INV prev. line

Approximate TSP

$\ell(u,v) \le \ell(u,x) + \ell(x,v)$

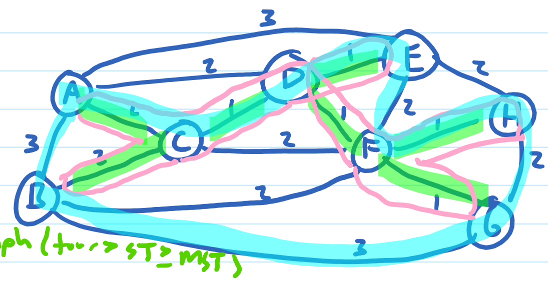
(FIND-TSP-METRIC)

FIND-TSP-Δ : Given weighted undirected G that satisfies triangle inequality, find lowest weight tour.

$S = A\,C\,B\,D\,E\,F\,G\,H\,A$

$S' = A\,C\,B\,\cancel{C}\,D\,E\,F\,G\,\cancel{F}\,H\,\cancel{F}\,\cancel{D}\,\cancel{C}\,A$

APPROX-FIND-TSP-Δ (G)
$T \leftarrow$ MST(G)
$S \leftarrow$ preorder(T)



$\ell(S') \underset{\#}{=} 2\,\ell(\boxed{T})$ S' goes over each edge twice

shortest tour $\ell(H^-) \underset{\#}{\ge} \ell(T)$ H^- - one edge spans the graph (tour > ST ≥ MST)

$\ell(S) \underset{\#}{\le} \ell(S')$ Δ inequality

$\ell(S) \le \ell(S') = 2 \cdot \ell(T) \le 2 \cdot \ell(H^-)$

So this is a 2-approximation of FIND-TSP-Δ

General TSP is not ~~2~~ k-approximable for $k \ge 1$ (unless P = NP)

HC(G)

1) Build complete G' with same vertices as G

Set $w(u,v) = \begin{cases} 1 & \text{if } (u,v) \text{ in } G \\ \cancel{2}\ \ 2+n \quad 2+(k-1)n & \text{otherwise} \end{cases}$

2) find a tour in G' with cost $\le$ ~~2~~ k $\cdot$ optimal

3) if cost of that tour is $\le$ ~~2n~~ kn, output YES
else output NO

if G has HC then G' has tour of cost n
so step 2 returns tour of cost $\le$ ~~2n~~ kn
alg outputs YES

if G has HC then G' has tour of cost n kn

so step 2 returns tour of cost $\leq 2n$

alg outputs YES

$\geq 2+(k-1)n+(n-1)$

$\geq kn+1$

if G has no HC then optimal tour in G' has cost $\geq 2+n+(n-1)$

$2n+1$

so step 2 returns tour of cost $\geq 2n+1$

so outputs NO

Randomized Algorithms

3-SAT-RANDOM

1) pick random assignment
2) while # clauses satisfied $\le \frac{7}{8}k$
3) pick a new random assignment

Z : # clauses satisfied by random assignment

$= Z_1 + Z_2 + \cdots + Z_k$ where $Z_i := \begin{cases} 1 & \text{if clause T} \\ 0 & \text{otherwise} \end{cases}$

$E[Z] = E[Z_1] + \cdots + E[Z_k]$

$= \frac{7}{8} + \cdots + \frac{7}{8} = \frac{7}{8}k$

$P(\text{random assignmen satisfies} \ge \frac{7}{8}k \text{ clauses}) \ge \frac{1}{8k}$

$E[\text{iterations of loop}] \le 8k$

3-SAT-RANDOM is expected poly-time alg to find assignment that satisfies $\ge \frac{7}{8}k$ clauses