Approximation for Maximum Acyclic Subset

https://en.wikipedia.org/wiki/Feedback_arc_set#Approximations

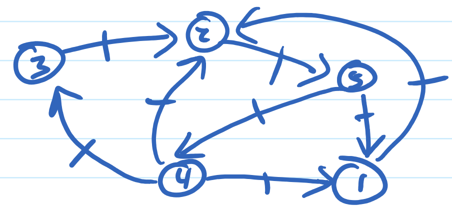
APPROX-MAX-ACYCLIC(G)

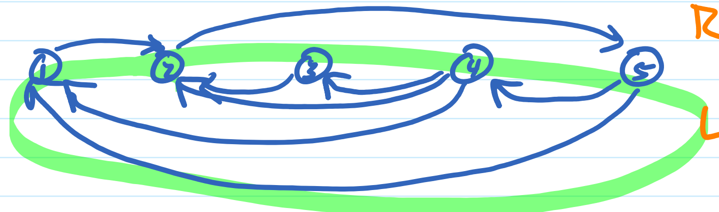
pick a random permutation of vertices
L ← edges (u,v) c(u) > c(v)
R ← edges (u,v) c(u) < c(v)
return the larger of L R

$|E| \geq |E^*|$

$\max(|L|, |R|) \geq \frac{1}{2}|E| \geq \frac{1}{2}|E^*|$

2-approximation





topo-sort: ordering of verts on a line so all edges ⟶

any acyclic graph has a topo-sort
(hr

P-Complete

Why don't we talk about P-complete?

→ X is P-complete if
1) $X \in P$
2) $Y \in P \rightarrow Y \leq_p X$

Thm For any $X, Y \in P$, $X \leq_p Y$

Let $X, Y \in P$. Then $\exists A, B$ that solve X, Y respectively in poly-time

For Karp reductions ($\leq_m$), $X \leq_m Y$ means

$\underline{X(x)}$
$y \leftarrow f(x)$ poly-time
return $Y(y)$

$\underline{X\text{-}V/A\text{-}Y(x)}$
~~$y \leftarrow B(\text{"Fred"})$~~
return $A(x)$

For any $X, Y \in P$ s.t. Y is non-trivial (not always yes or always no)
$X \leq_m Y$

Find y_{yes} s.t. $Y(y_{yes}) = Y$
y_{no} s.t. $Y(y_{no}) = N$

$\underline{X\text{-}KARP\text{-}Y(x)}$
if $A(x) = Y$
  $y \leftarrow y_{yes}$
else
  $y \leftarrow y_{no}$
return $B(y)$

$f(x)$

COR: for any $X \in P$, X is P-complete

COR: If $P = NP$ then any $X \in P$ is NP-complete

Let $X \in P$
Then $X \in NP$ ✓
Let $Y \in NP$
So $Y \in P$
and $Y \leq_p X$
X is NP-complete

for all $Y \in NP$, ✓
$Y \leq_p X$

Y is NP-complete and $Y \in P$ Then $P = NP$

$P \subseteq NP \subseteq PSPACE \subseteq EXPTIME$

THM: Suppose Y is $\underbrace{\text{PSPACE-complete}}_{\text{defined using Karp } \le_m}$ and $Y \in NP$. Then PSPACE = NP

Let Y be PSPACE-complete [want PSPACE $\subseteq$ NP since we have NP $\subseteq$ PSPACE; together NP = PSPACE]

Let $X \in$ PSPACE [want $X \in NP$; in other words there is a poly-time verifier for X]

Then $X \le_m Y$

Also $Y \in NP$ so $\exists$ poly-time verifier Y-VERIFY(y, z)

Consider X-VERIFY(x, z)
compute $y \leftarrow f(x)$ using the f from $X \le_m Y$
return Y-VERIFY(y, z)

$X(x) = YES \rightarrow Y(f(x)) = YES \rightarrow \exists z$ s.t. $|z|$ is poly in $|f(x)|$ and Y-VERIFY$(f(x), z) = YES$
(find that z)
$|f(x)|$ is poly in size of x
so $|z|$ is poly in size of x
X-VERIFY$(x, z) = YES$
so $\exists z$ s.t. $|z|$ is poly in $|x|$ and X-VERIFY$(x, z) = Y$

$X(x) = NO \rightarrow Y(f(x)) = NO \rightarrow$ no z s.t. Y-VERIFY$(f(x), z) = YES$
so no z s.t. X-VERIFY$(x, z) = YES$

so X-VERIFY is a poly-time verifier for X

so $X \in NP$

ZPP and NP

ZPP: decision problems with expected poly-time algorithms

NP: decision problems with poly-time verification algorithms

ZPP $\subseteq$ NP: Let $X \in$ ZPP and find RAND-X that solves X in expected poly-time ($p(|x|)$ for some poly p)

Write <u>X-VERIFY(x, y)</u> cut off if no more bits available in y or takes time longer than $p(|x|)$

return RAND-X(x) using y as source of random bits

X(x) = YES → RAND-X(x) = YES in expected time $p(|x|)$ for some poly p
→ some execution path of RAND-X(x) runs in time $\le p(|x|)$
→ that path consumes $\le p(|x|)$ random bits
let y that sequence of random bits (so $|y|$ is poly in $|x|$)
X-VERIFY(x, y) = YES and runs in time $\le p(|x|)$

X(x) = NO let y be any random bits RAND-X(x) = NO on all paths
X-VERIFY(x, y) = NO

RP: problems w/ poly-time algorithm w/ one-sided error (YES → p(YES) > c for some c > 0; NO → p(NO) = 1)

co-RP

ZPP $\subseteq$ RP

ZPP = co-ZPP

ZPP $\subseteq$ co-RP

→ ZPP $\subseteq$ RP $\cap$ co-RP

ZPP = RP $\cap$ co-RP

Undecidable Problems

HALT : given ~~algorithm~~ program A and input x , does A(x) stop? (halt)

undecidable

TOT : given ~~algorithm~~ program A, does it stop on all inputs?

undecidable

HALT $\underbrace{\le_T}_{\text{Turing reduction}}$ TOT

" Cook reduction w/o poly-time restriction

HALT(A, x)

Write program B(y)

return A(x)

return TOT(B)