Coping With NP- Completeness

Special Cases

Average Case

Approximation Algorithms

Intelligent Brute Force

Heuristics

Special Cases

2-CNF Sat:

Use resolution:

$$(x \vee y \vee z) \wedge (\bar{z} \vee w \vee \bar{v})$$

resolve to

$$(x \vee y \vee w \vee \bar{v})$$

Sat iff $\square$ (the empty clause)

<u>cannot</u> be obtained by resolution

Eliminate one variable at a time by resolving it in all possible ways

For 3-Sat, or general sat, clauses can get arbitrarily long: 2^{2n}, 3^n

2-sat

$(x \vee \bar{z}) \wedge (z \vee y)$ gives $(x \vee y)$

Resolution preserves 2-sat

$O(n^2)$ possible clauses

$O(n^3)$ time

Like transitive closure,

all-pairs shortest paths

Faster: Formula $\rightarrow$ Graph

$(x \vee y) \Rightarrow \bar{x} \rightarrow y$ (edges)

$x \leftarrow \bar{y}$

Literals are vertices

Satisfiable iff no literal and its negation are in the same strong component. (Why?)

$O(m+n)$ where $m =$ # clauses

$n =$ # literals

(Can propagate single-literal clauses first, or use $x \equiv x \vee x \Rightarrow \bar{x} \rightarrow x$)

Special cases

Min vertex cover for a bipartite graph

Find a maximum matching.

Search for augmenting paths from free vertices on A side. Let reached vertices be S, others $\bar{S}$.

Let $C = (B \cap S) \cup (A \cap \bar{S})$

This is a minimum vertex cover.

$B \cap S$: all matched in S $\qquad \bar{S}$

B

A

S

$A \cap \bar{S}$:

all matched in $\bar{S}$

No matched $B \cap S$ to $A \cap \bar{S}$ edges

No unmatched $A \cap S$ to $B \cap \bar{S}$ edges

$\Rightarrow (B \cap S) \cup (A \cap \bar{S})$ is a vertex cover

of size = maximum matching

$\Rightarrow$ minimum (every edge of a matching

must be covered)

Approximation

General vertex cover

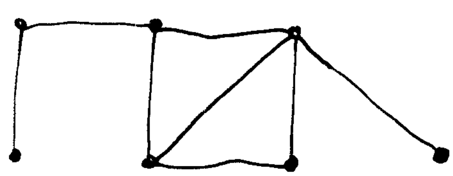
Find a maximal matching (no new edges can be added)

C = both ends of all matched edges

covers since maximal matching

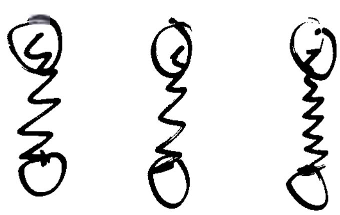
2- approximation

O(m)- time



[illegible]

[illegible]



Approximation

Minimum tour (TSP) with Δ inequality:

$$d(x,y) + d(y,z) \geq d(x,z)$$

$\Rightarrow$ given any tour with repeats, can find a tour no longer by dropping repeats

Find a minimum spanning tree, build a tour as a depth-first traversal (each edge used twice), delete repeated vertices.

2-approximation

1.5 approximation

Find an MST T

\# odd-degree vertices is even

Find a min-cost perfect matching on

odd degree vertices P

$T \cup P$ has all vertices of even degree:

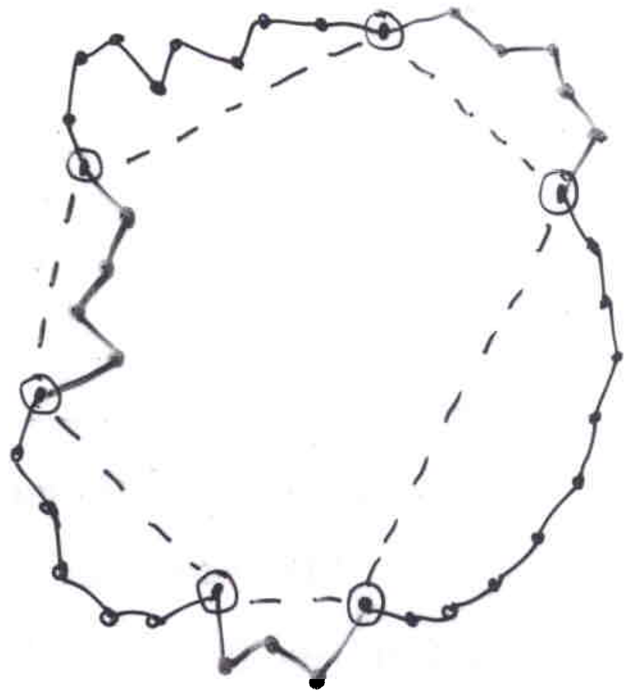
Find an Eulerian tour, delete repeated

vertices.

If R is a min-cost tour $|T| \le |R|$,

$|P| \le |R|/2 \Rightarrow |T+P| \le 1.5\,|R|$

($|\ |$ denotes cost)



R decomposes into two ways of pairing odd-degree vertices, gives to matchings of odd-degree vertices.

(Subset Sum) Knapsack Problem

$0 \le x_1, x_2, \ldots x_n$ Subset sums to k.

L = all possible sums $\le k$ of first j numbers.

$O(kn)$ - time alg.

$L_0 = \{0\}$

$L_{j-1} + x_j = L'_j$

Merge L_{j-1} and L'_j, dropping duplicates

$= L_j$

Extend to approx:

get a sum $\le k$, but max (within $1+\epsilon$ factor)

sum $\geq \frac{k}{1+\epsilon}$.

Tiply merging alg, but drop #'s within a factor of $1+\frac{\varepsilon}{2n}$ of each other.

$L_j \oplus L_j' \Rightarrow$

add next elmt to L_j only if $> 1+\frac{\varepsilon}{2n}$ of prev. added elmt.