Coping With NP-Completeness

Special Cases
Average Case
Approximation Algorithms
Intelligent Brute Force
Heuristics
Special Cases

2-CNF Sat:

Use resolution:

\[(x \lor y \lor z) \land (\overline{z} \lor w \lor \overline{v})\]

resolve to

\[(x \lor y \lor w \lor \overline{v})\]

Sat iff \[\square\] (the empty clause)

cannot 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: \[2^n\]
2-sat

\((x \lor \bar{z}) \land (\bar{z} \lor y)\) gives \((x \lor y)\)

Resolution preserves 2-sat

\(O(n^2)\) possible clauses

\(O(n^3)\) time

Like transitive closure,

all-pairs shortest paths
Faster: $\text{Formula} \rightarrow \text{Graph}$

$(x \lor y) \Rightarrow \overline{x} \rightarrow y$ (edges)

$x \leftarrow \overline{y}$

Literals are vertices

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

$O(m+n)$ where $m = \#\text{clauses}$

$n = \#\text{literals}$

(Con propagate single-literal clauses first,
or use $x = x \lor x \Rightarrow \overline{x} \rightarrow x$)
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.
\[ B_n \setminus S : \text{all matched in } S \]

\[ \overline{S} \]

\[ \text{No matched } B_n \setminus S \text{ to } \overline{A \cap \overline{S}} \text{ edges} \]

\[ \text{No unmatched } A \cap \overline{S} \text{ to } B_n \setminus S \text{ edges} \]

\[ \Rightarrow (B_n \setminus S) \cup (A \cap \overline{S}) \text{ is a vertex cover of size } \leq \text{ maximum matching} \]

\[ \Rightarrow \text{ minimum (every edge of a matching must be covered)} \]
Approximation

General vertex cover

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

$G$: both ends of all matched edges covers since maximal matching

$2$-approximation

$O(m)$-time
Approximation

Minimum tour (TSP) with $\Delta$

inequality:

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

⇒ 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 $|T| \leq |R|$, $|P| \leq |R|/2 \Rightarrow |T+P| \leq 1.5|R|$
R decomposes into two ways of pairing odd-degree vertices, gives to matchings of odd-degree vertices.
(Subset sum) Knapsack Problem

\[ 0 \leq x_1, x_2, \ldots, x_n \text{ Subset sums to } k. \]

\[ L = \text{all possible sums } \leq k \text{ of first } j \text{ numbers.} \]

\( O(nk) \) - time alg.

\[ L = \emptyset \]

\[ L_{j-1} + x_j = L_j \]

Merge \( L_{j-1} \) and \( L_j \), dropping duplicate entries.

\[ = L_j \]

Extend to approx:

get a sum \( \leq k \), let \( \text{max} \) (within \( 1 + \varepsilon \) factor)

\[ \text{sum} \geq \frac{k}{1+\varepsilon}. \]
freely merging alg, but drop $N$ within a factor of $1 + \frac{\epsilon}{2\text{ne}}$ of each other.

$\overline{L}_j \otimes \overline{L}_i \rightarrow$

add next elt to $L_j$ only if $\frac{\text{prod. added elt}}{2\text{ne}} > 1 + \frac{\epsilon}{2\text{ne}}$