

Lec 12. Beyond polynomial-time algorithm

Nikolaos Melissinos

UNLIKELY TO BE P-TIME, WHAT NEXT?

POPULAR ALGORITHMIC APPROACHES

- Exponential running time, but as fast as possible.
↪ Exponential-time algorithm. in time $f(n)$ for as small f as possible.
- Not exact/optimal solution, but as close to optimal as possible.
↪ Approximation algorithm. runs in P-time.
- Exponential running time, but exponential w.r.t a small parameter k .
↪ Fixed-parameter algorithm. runs in time $f(k) \cdot n^{O(1)}$.
- Produced intended solution with high probability.
↪ Randomized algorithm. can be optimal or approximate.

DIRECTED HAMILTONIAN PATH

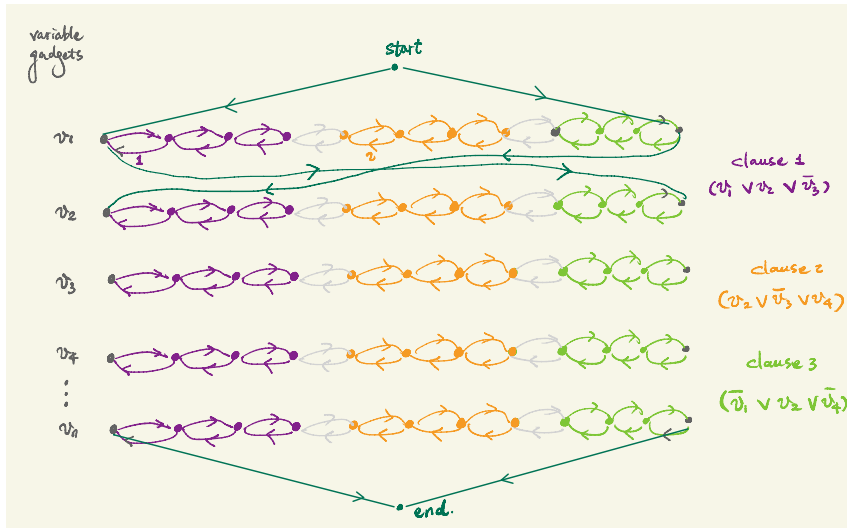
A Hamiltonian path of a directed graph $G = (V, E)$ is a directed path which visits every vertex of G precisely once.

PROBLEM DIRECTED HAMILTONIAN PATH

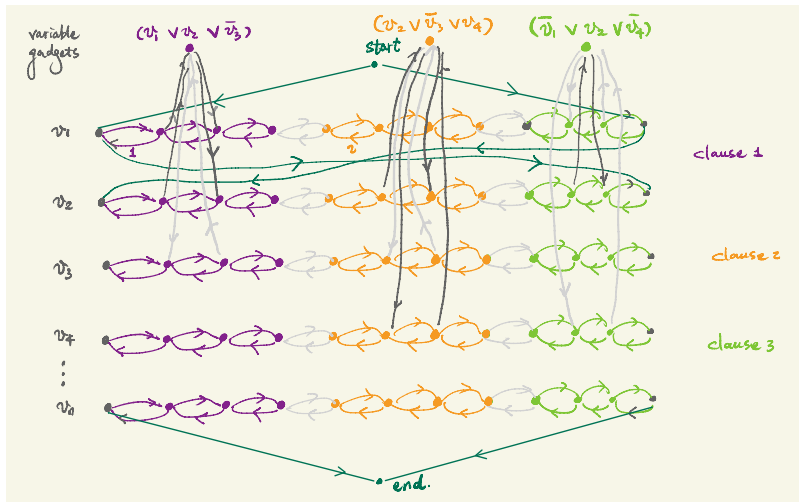
INPUT a graph G .

QUESTION does G have a Hamiltonian path?

REDUCTION: 3SAT TO DIRECTED HAMILTONIAN PATH



REDUCTION: 3SAT TO DIRECTED HAMILTONIAN PATH



DYNAMIC PROGRAMMING FOR HAM

Naive algorithm: guess all possible tours, $n!$ many of them.

Dynamic programming algorithm in $2^n \cdot n^{O(1)}$ time.

For every vertex subset X and $v \in X$,

$P[X, v] = 1$ if and only if there exists a Hamiltonian path of $G[X]$ ending in v .

- 1 Initialize: $P[\{w\}, w] = 1$ for every $w \in V$.
- 2 Inductive step: assume $P[X, v]$ is known for all $|X|$ of size i and $v \in X$. Let's compute $P[X, v]$ for each $|X|$ of size $i + 1$ and $v \in X$.

$$P[X, v] = \bigvee_{w \in X \setminus v} P[X \setminus v, w] \cdot [(w, v) \in E(G)].$$

- 3 There exists a Hamiltonian path of G if and only if a vertex v s.t. $P[V, v] = 1$.

VERTEX COVER

A vertex cover of a graph is a vertex subset X of G such that X takes at least one endpoint of every edge in G .

VERTEX COVER

INPUT a graph G and an integer k .

QUESTION does G have a vertex cover of size at most k ?

VERTEX COVER is NP-complete; reduction from INDEPENDENT SET.

APPROXIMATION FOR VERTEX COVER

GREEDY ALGORITHM FOR VERTEX COVER

- 1 Greedily find a maximal matching M .
- 2 Output $V(M)$.

- The output of the algorithm is indeed a vertex cover. (Why?)
- Analysis:

$$|M| \leq \text{opt vc} \leq \text{output vc} = 2 \cdot |M|.$$

Therefore, $\text{output vc} \leq 2 \cdot \text{opt vc}$.

- 2-approximation: algorithm outputs a solution no larger than twice the optimal.

PARAMETERIZED VERTEX COVER

PARAMETERIZED VERTEX COVER

INPUT a graph G and an integer k .

PARAMETER k .

QUESTION does G have a vertex cover of size at most k ?

FPT-ALGORITHM FOR VERTEX COVER

Algorithm **VC**(G, k)

- 1 If G has no edge **return** YES.
- 2 Else if $k = 0$ **return** NO.
- 3 Else (comment: G has an edge and $k > 0$)
 - 1 Pick an edge uv .
 - 2 Return **VC**($G - u, k - 1$) or **VC**($G - v, k - 1$).

FPT-ALGORITHM FOR VERTEX COVER

Runtime analysis:

- Recursive calls of $\mathbf{VC}(G, k)$ can be expressed as a branching tree \mathcal{T} .
- Parameter k strictly decreases by depth \rightsquigarrow depth at most k .
- \mathcal{T} has at most 2^k leaves \rightsquigarrow runs in time $2^k \cdot \text{poly}(n)$.

RANDOMIZED APPROXIMATION FOR MAX CUT

MAX CUT

INPUT a graph $G = (V, E)$.

QUESTION find a bipartition of V into V_1, V_2 maximizing the crossing edges between V_1 and V_2 .

MAX CUT is NP-hard; reduction from 3SAT via a series of reductions.

RANDOMIZED APPROXIMATION FOR MAX CUT

RANDOM GREEDY ALGORITHM FOR MAX CUT

- 1 For each vertex v , decide where to put v with prob 0.5, independently at random.
 - 2 Count the crossing edges between V_1 and V_2 .
- For arbitrary edge e , probability that e counts as 'crossing' is $1/2$.
 - $E(\# \text{crossing edges}) = \sum_{e \in E} P(e \text{ is crossing}) = |E|/2$.
 - In **expectation**, the output is at least half the optimal solution.
 - Repeat the greedy algorithm several times \rightsquigarrow turn the 'expectation' into probability guarantee.
 - Derandomization techniques.