Lecture 14

N-V algo for TSP
- $O(n^3 \phi)$
- in expectation.

Core Algorithms

$1, \ldots n$ $\quad (x_1, \ldots x_n) \in \{0,1\}^n$

$p_1 \ldots p_n$
$w_1 \ldots w_n$.

maximize $p_1 x_1 + \cdots + p_n x_n$

subject to $w_1 x_1 + \cdots + w_n x_n \le W$
$x_i \in \{0,1\} \; \forall i$

Fractional Knapsack
→ where fractions of items can be stored

Relaxation:
$x_i \in [0,1] \; \forall i$

ie x_i - allowed to be a real value.

Greedy for frac. Knapsack

Sort items in the descending order of p_i/w_i ratio

$1 \quad 2 \quad \ldots \quad n$

$p_i/w_i \ge \frac{p_{i+1}}{w_{i+1}}$

Add items as long as weight does not exceed capicity.

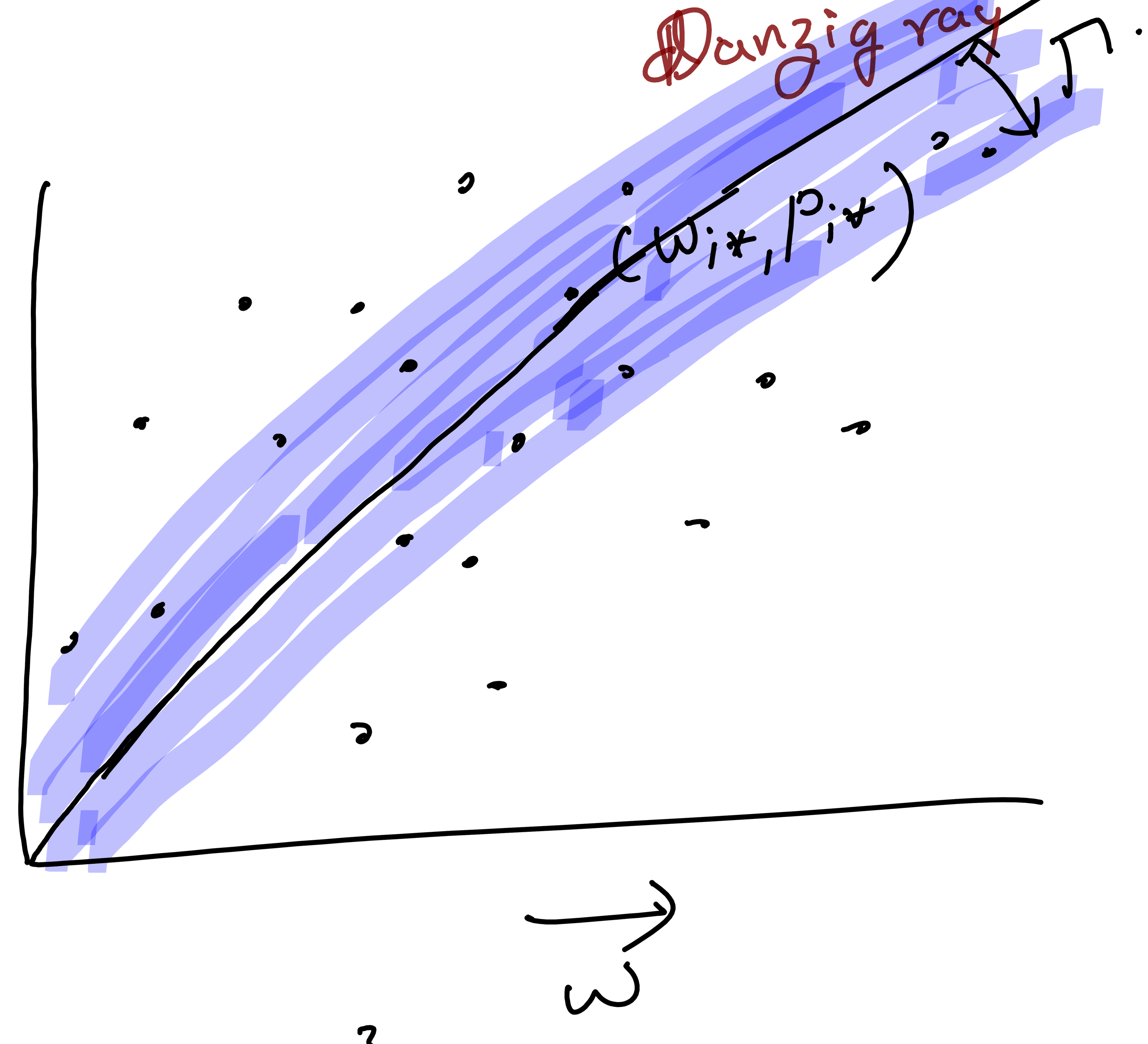
$i_* - 1$

$x_1 \quad x_2 \quad \ldots \quad x_{i^*-1}$

obs:- The solution is optimal & there is at one most one item i^* s.t $x_{i^*} \notin \{0,1\}$

(w_{i^*}, p_{i^*}).

$\hat{x}$ be an optimal solution.



Stmt: if $w_1 \cdots w_n$ & $p_1 \cdot\cdot p_n$ are ϕ-perturbed then $\exists\ \Pi$ s.t # of items within a distance of Π from the Danzig ray is $O(n^{\frac{1}{3}})$

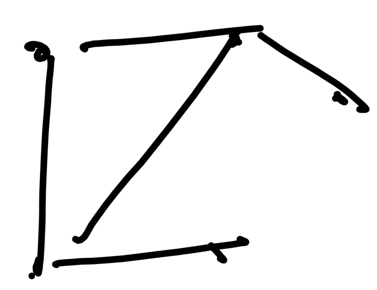
$\therefore$ Expected running time = Sort(n) + $(n^{\frac{1}{3}})^3 \cdot \phi$.

$\simeq O(n \log n)$.

Details
See Röglin's notes.

A quick review of Complexity classes

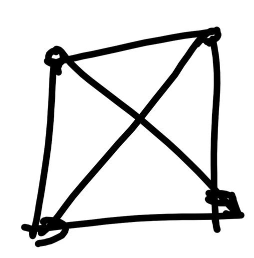
NP: non-deterministic polynomial time.
set of all problems that have efficiently verifiable soln/witnesses.

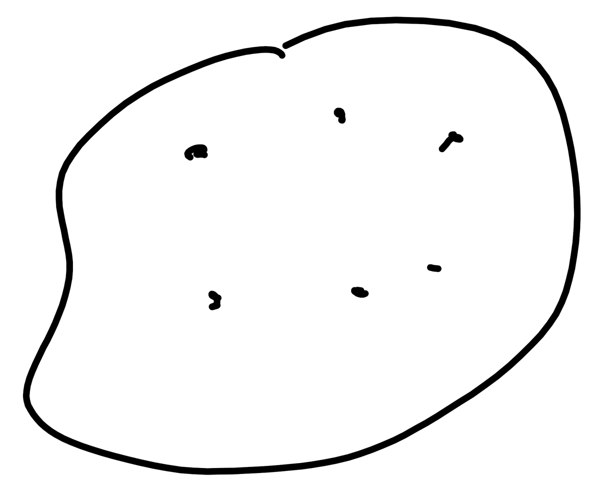


e.g. CLIQUE problem.

i/p = $G = (V, E)$ & $k > 0$

o/p: YES iff G has a clique on k vertices





witness for YES instance

$\boxed{\text{A subset } S \subseteq V,\ |S| = k \text{ s.t } \forall\, i \neq j \in S,\ (i,j) \in E.}$

Vertex Cover
Dominating set
Hamiltonian cycle

Given G & K is there a path of length K

$\therefore$ CLIQUE $\in$ NP

NP-hard

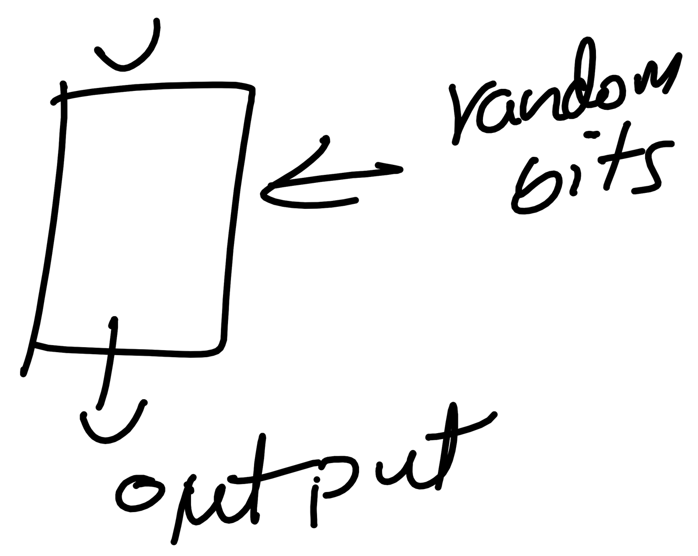
A problem B is NP hard if:
there is a poly time algo for B then for every problem $A \in NP$, we have a polynomial time algorithm.

Note:- CLIQUE, DomSet, VC, HamCycle, k-path, k-cycle
Knapsack, TSP, clustering.
All are NP hard problem. (in fact NP-complete).

For us:- [P≠NP hypothesis]
No NP-complete problem is likely to have a polynomial time algorithm. ie $NP \neq P$.

TSP-implementations
Concorde-

Suppose an algorithm A used random coin tosses.



Monte Carlo & Las Vegas.

Monte Carlo:
runtime expectation.
o/p: guarantee

Las Vegas:
run time guaranteed
o/p: expectation (ie possibility of errors).

Definition A problem B is said to be in ZPP (zero error prob. poly time)
if it has a randomized algo which has expected poly time & outputs correct value.