Advanced topics on Algorithms

Luciano Gualà
www.mat.uniroma2.it/~guala/

Beyond Worst-Case Analysis
Episode IV
(the final one)

Robust distributional analysis

cost(A,z): amount of resources algorithm A consumes for input z

- e.g., running time, space, I/O operations, cost of a solution

Worst-case analysis:

$$\max_{z} \text{cost}(A,z)$$

Average-case analysis (w.r.t. a distribution D over inputs):

$$E_{z\sim D}[\text{cost}(A,z)]$$

Issues of average-case analysis:

- uncertainty about D
- overfitting algorithm to D

goal: can we find a sweet spot in between?

Smoothed analysis

- an adversary picks an input
- nature slightly perturbs it (add a small random perturbation)

When useful?

- bad inputs should be "fragile"
- usually for running time

Smoothed complexity:

$$\sup_{z} \; E_{r(\sigma)} [\mathrm{cost}(A, z+r(\sigma))]$$

$\sigma = 0$ ⇒ worst-case analysis

$\sigma \gg 0$ ⇒ ≈ average-case analysis

$r(\sigma)$: perturbation
σ : size of the perturbation

goal: prove a good complexity as function of n and $1/\sigma$

Smoothed Analysis of Pareto Curves

the knapsack problem

Input:

- n objects
- object i has value v_i and weight w_i
- a knapsack capacity W

Feasible solution:

a subset $S \subseteq \{1,2,\ldots,n\}$ of the objects such that $\sum_{i \in S} w_i \leq W$

measure (max):

value of S : $\sum_{i \in S} v_i$

Definition

A solution S dominates a solution T if:

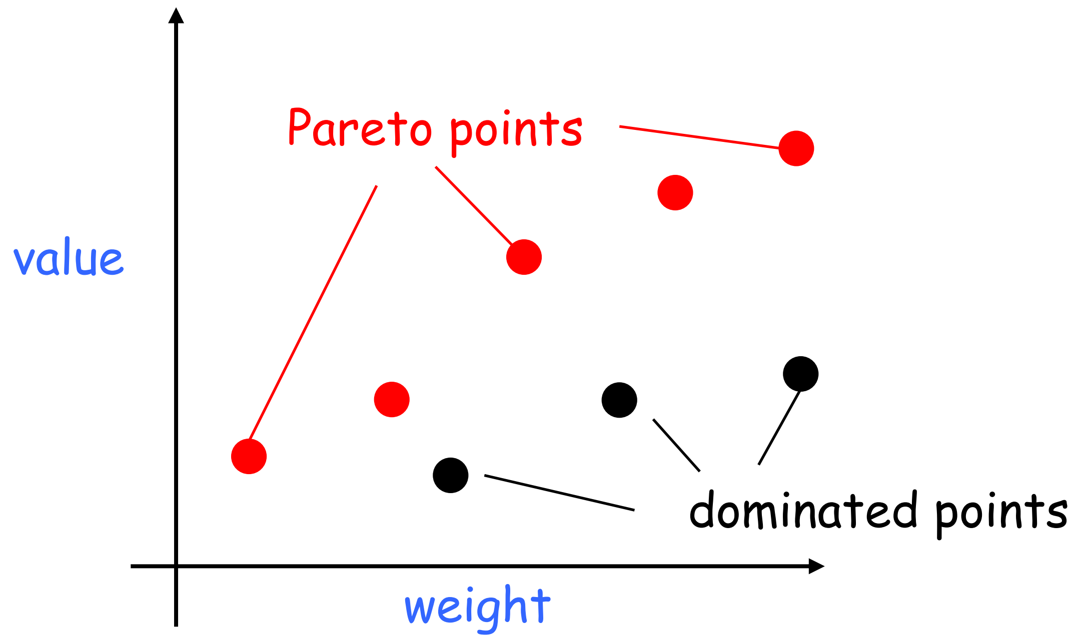
(i) the total value of S is at least the value of T;

(ii) the total size of S is at most the size of T;

(iii) at least one of these two inequalities is strict.

Obs: S renders T moot (T can be safely pruned without regret)

Pareto Curve of a Knapsack instance: set of all undominated solutions



A Knapsack algorithm:

1. Generate the Pareto curve.
   (if multiple solutions have identical total value and total weight, an arbitrary one of them is retained.)
2. Among all solutions in the Pareto curve with total weight at most the knapsack capacity W, return the one with the largest total value.

PC_i: Pareto curve of the first i items

PC_0 = empty set

PC_i can be computed from PC_{i-1} in time $O(|PC_i|)$

Exercise: figure it out

running time: $O(\sum_i |PC_i|) = n \min\{2^n, n\, w_{max}, n\, v_{max}\}$

if weights/value are integrals

fact: The size of the Pareto curve can be exponential in the worst case

Theorem [Beier & Vocking, 2006]
In smoothed Knapsack instances (see next slide), the expected size of the Pareto curve is $O(n^2/\sigma)$, where σ is a measure of "perturbation size".

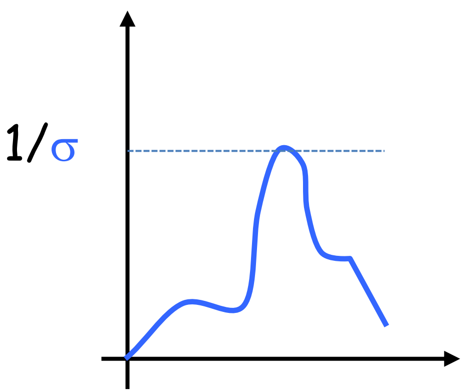
⇒ the Knapsack algorithm has expected running time of $O(n^3/\sigma)$
(polynomial if $1/\sigma$ is bounded by a polynomial function of n)

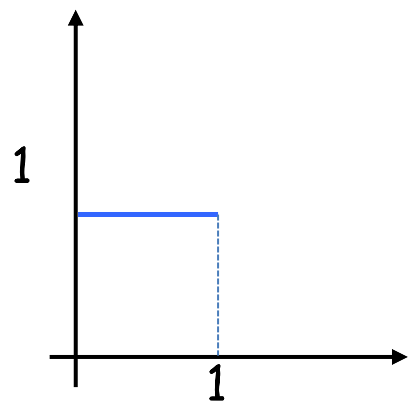
model of perturbation

W.l.o.g. assume values and weights are in [0,1]

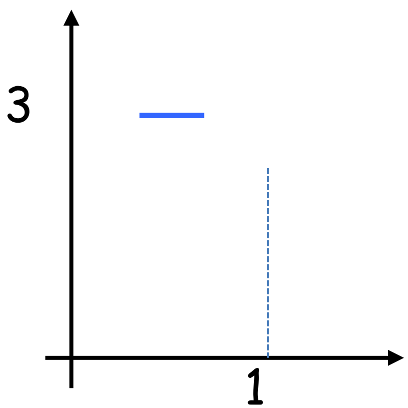
- values v_i are adversarial
- each weigh w_i is drawn independently according to a density function f_i: $[0,1] \rightarrow [0, 1/\sigma]$ which is not "too spiky"

  $\sigma \in (0,1]$ is a measure of "perturbation size"





weight uniform on [0,1]



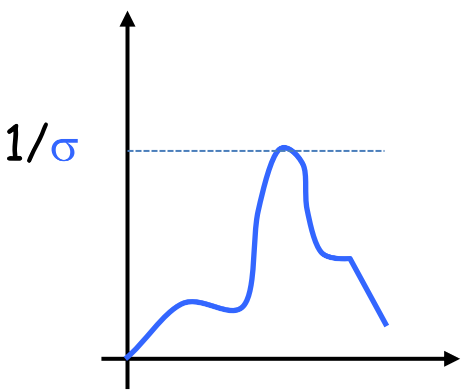
weight uniform on [1/3,2/3]

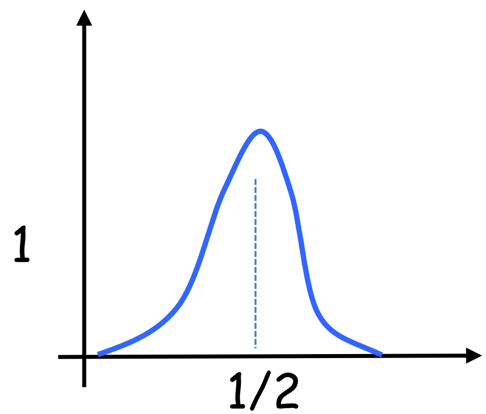
model of perturbation

W.l.o.g. assume values and weights are in [0,1]

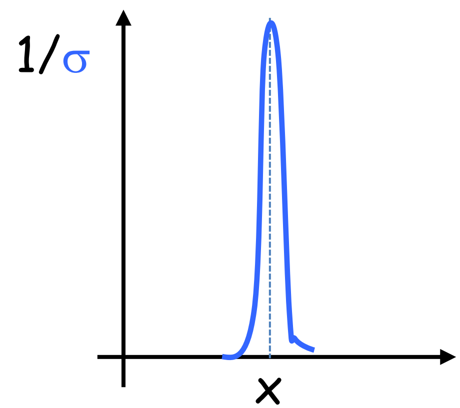
- values v_i are adversarial
- each weigh w_i is drawn independently according to a density function f_i: $[0,1] \rightarrow [0, 1/\sigma]$ which is not "too spiky"

  $\sigma \in (0,1]$ is a measure of "perturbation size"





weight $\frac{1}{2}$+ Gaussian perturbation of mean 0 and variance σ^2



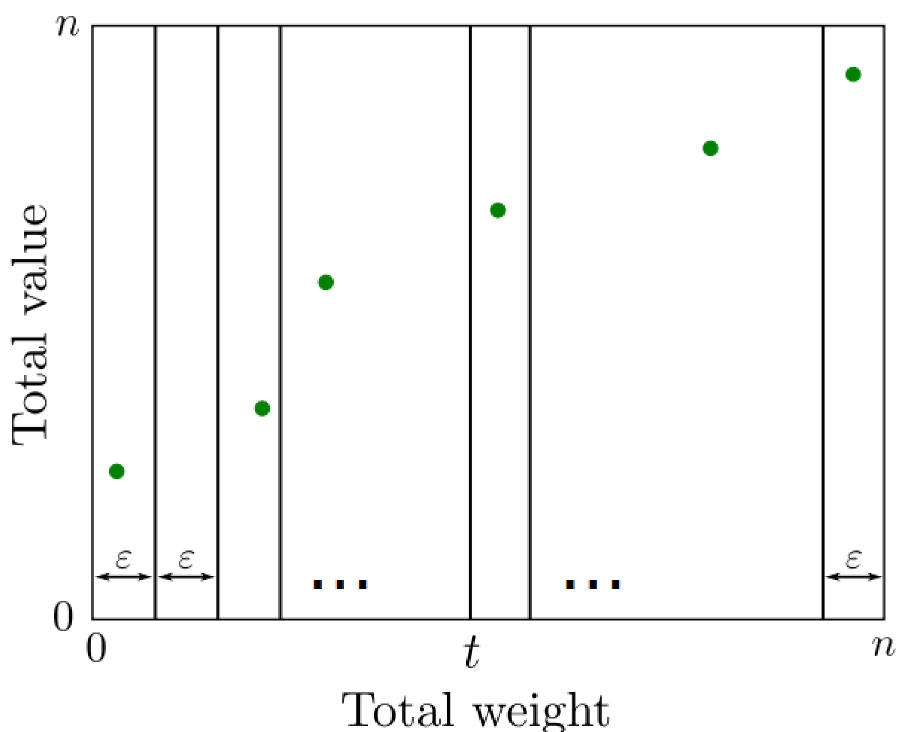
weight with value concentrated around x

N: number of Pareto points

goal: bound the expected value of N

idea: decompose N into simpler random variables that are easier to relate to the hypothesis that the item weights are random

- each solution is a point in the [0,n]x[0,n] box
- decompose the box into n/ε vertical "slices" each of width ε
- ε is a parameter internal to the analysis (it will not appear in the final bound)
- choose ε such that probability of having two points in the same slice ≈ 0 (e.g., inverse doubly exponential in n)



warm-up analysis

P_t: max-value solution with weight at most t (t: slice boundary)
V_t: value of P_t

S_t: min-weight solution with value greater than V_t

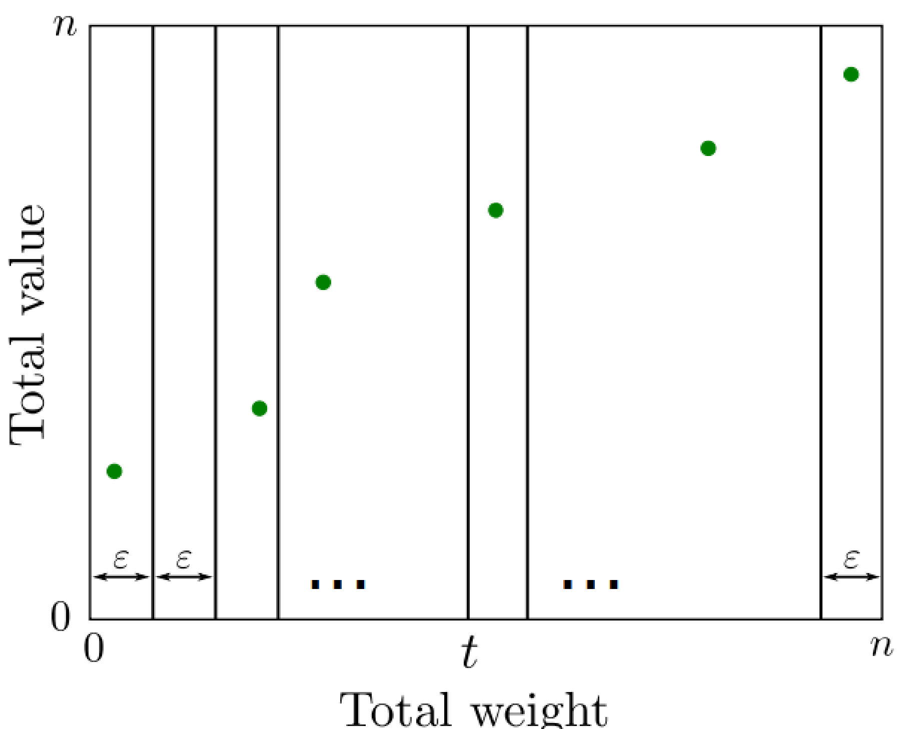
N_t: indicator r.v. =1 if weight of $S_t \in (t, t+\varepsilon)$

claim: $\sum_t N_t \geq N$ (with probability 1)

fix a Pareto point S
t= slice boundary to the left of S

➡ value of S > V_t

➡ N_t =1



analysis

P_{ti}: max-value solution not including item i with weight at most t
V_{ti}: value of P_{ti}

S_{ti}: min-weight solution including item i with value greater than V_{ti}

N_{ti}: indicator r.v. =1 if weight of $S_{ti} \in (t, t+\varepsilon)$

claim: $\sum_t \sum_i N_{ti} \geq N$ (with probability 1)

proof

fix a Pareto point S
t= slice boundary to the left of S

value of S > value of P_t

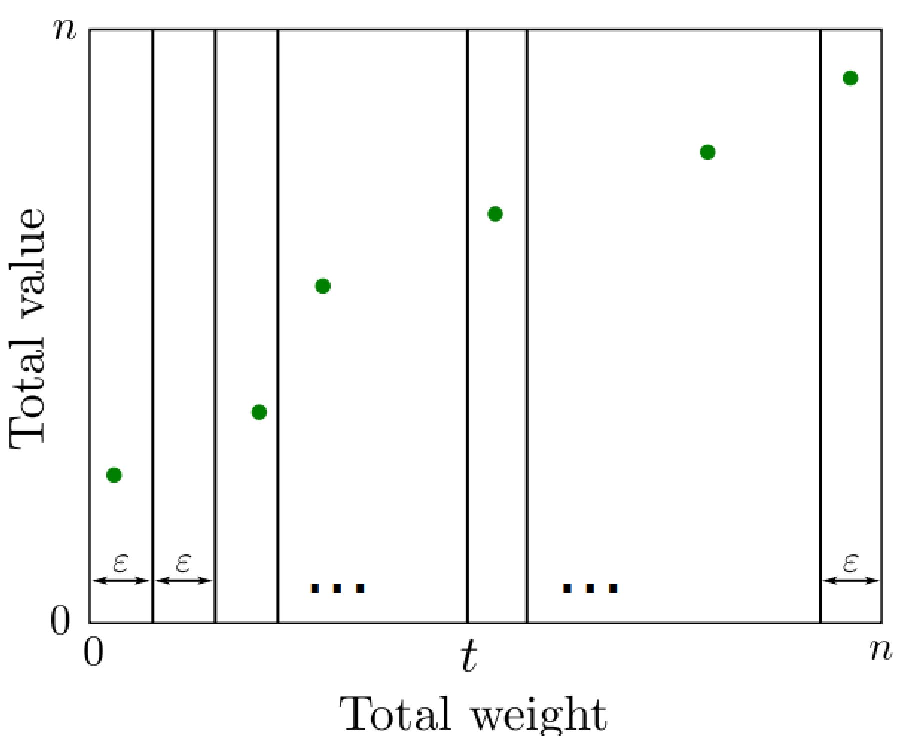
⇒ there is $i \in S \setminus P_t$

⇒ $P_{ti} = P_t$

⇒ $S = S_{ti}$

⇒ value of S > V_{ti}

⇒ $N_{ti} = 1$ ■



analysis

goal: $E[N] \leq n^2/\sigma$

claim: $\sum_t \sum_i N_{ti} \geq N$

it suffices to show that:

Lemma: $E[N_{ti}] \leq \varepsilon/\sigma$ for each i,t

proof

fix i and t, and condition on all of the random weights except for w_i

P_{ti} and hence V_{ti} are now fixed

consider all the set including i and order them by total size

such relative ordering is now fixed

⇒ S_{ti} is fixed and its (random) weight is $W+w_i$ for some fixed W

$E[N_{ti}] = Pr[N_{ti}=1] = Pr[\text{weight of } S_{ti} \in (t,t+\varepsilon)] \leq \varepsilon/\sigma$ (since $f_i(x) \leq 1/\sigma$)

■