Algorithms: Design and Analysis, Part II

Approximation Algorithms for NP-Complete Problems

A Greedy Knapsack Heuristic

Strategies for NP-Complete Problems

(1) identify computationally tractable special cases

example: knapsack instances with small capacity

[i.e., knapsack capacity W = polynomial in number of items n]

(2) heuristics — today

- pretty good greedy heuristic
- excellent dynamic programming heuristic

} for knapsack

(3) exponential time but better than brute-force search

example: $O(nW)$-time dynamic programming vs. $O(2^n)$ brute-force search

Ideally: should provide a performance guarantee (i.e., "almost correct") for all (or at least many) instances.

Knapsack Revisited

Input: n items. Each has a positive value v_i and a size w_i. Also, knapsack capacity is W

Output: a subset $S \subseteq \{1, 2, 3, \dots, n\}$ that

maximizes $\sum_{i \in S} v_i$

subject to $\sum_{i \in S} w_i \leq W$.

Tim Roughgarden

A Greedy Heuristic

Motivation: ideal items have big value, small size.

Step 1: Sort and reindex item so that

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \frac{v_3}{w_3} \geq \cdots \geq \frac{v_n}{w_n}$$

[i.e., nondecreasing "bang-per-buck"]

Step 2: pack items in this order until one doesn't fit, then halt.

Example: $v_1 = 2$ $w_1 = 1$

$W = 5$ $v_2 = 4$ $w_2 = 3$

$v_3 = 3$ $v_3 = 3$

$\Rightarrow$ greedy gives $\{1, 2\}$

[also optimal]

Quiz

Consider a knapsack instance with $v_1 = 2$ $w_1 = 1$

$v_2 = 1000$ $w_2 = 1000$

$W = 1000.$

Question: What is the value of the greedy solution and the optimal solution, respectively?

(A) 2 and 1000

(B) 2 and 1002

(C) 1000 and 1002

(D) 1002 and 1002

A Refined Greedy Heursitic

Upshot: greedy solution can be arbitrarily bad relative to an optimal solution.

Fix: add

Step 3: return either the Step-2 solution, or the maximum valuable item, whichever is better.

Theorem: value of the 3-step greedy solution is always $\geq 50\% \cdot$ value of an optimal solution.

[also runs in $O(n \log n)$ time]

[i.e., a "$\frac{1}{2}$-approximation algorithm"]