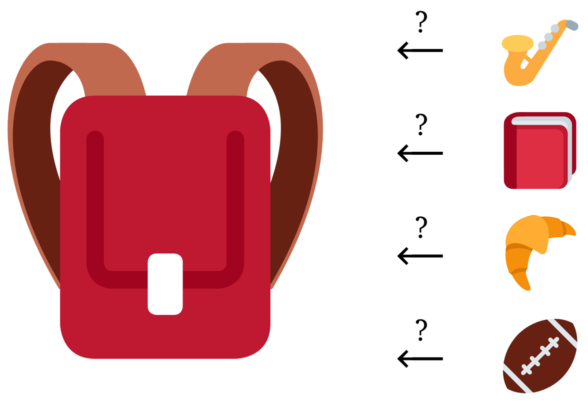
KNAPSACK AND SUBSET SUM WITH SMALL ITEMS

Adam Polak, Lars Rohwedder, Karol Węgrzycki

Max item size
$s = \max_i s_i$

KNAPSACK PROBLEM



Given:
- n items, i-th with *value* v_i and *size* $s_i \in \mathbb{Z}_+$
- knapsack capacity t

Find: a subset of items with
- total size not exceeding t
- maximizing total value

PSEUDOPOLYNOMIAL TIME ALGORITHMS

$O(nt)$ [Bellmann, 1957]

$O(n+st)$ [Kellerer and Pferschy, 2004]

$O(n^{2-\varepsilon})$ impossible* when $s, t \in \tilde{\Theta}(n)$ [Cygan et al., 2017]
* unless min-plus convolution hypothesis fails [Künnemann et al., 2017]

Can we get $O(n + \mathrm{poly}(s))$ time?

OUR RESULT

$O(n+s^3)$ time algorithm for Knapsack

Works for the more general variant with (binary encoded) multiplicities:
- multiplicities succinctly denote how many times each item appears in the instance;
- n denotes the (possibly much smaller) number of distinct items.

INGREDIENT #1: MAXIMAL PREFIX SOLUTIONS

1. Arrange items in descending order of efficiency v_i/s_i
2. Take maximal prefix with total size $\leqslant t$

Alternative definition: round down vertex solution to LP relaxation

Theorem: [Eisenbrand and Weismantel, 2018]
There is optimal solution that differs from maximal prefix solution by at most $2s$ items

INGREDIENT #2: CONVEX MAX-PLUS CONVOLUTION

Recall Bellman's dynamic program for Knapsack:

$DP_A[i] =$ max total value of items from set A of total size $\leqslant i$

Let $A_\ell =$ items of size exactly ℓ

$$\underbrace{DP_{A_1 \cup A_2 \cup \cdots \cup A_s}}_{= DP_A} = DP_{A_1} \oplus (DP_{A_2} \oplus (\cdots \oplus \underbrace{(DP_{A_{s-1}} \oplus DP_{A_s}}_{\text{max-plus convolution, i.e., } (u \oplus v)[k] = \max_i (u[i]+v[k-i])}) \cdots))$$

Two key observations: [Kellerer and Pferschy, 2004]
1. Each DP_{A_ℓ} is concave
2. Max-plus convolution of a concave vector and an arbitrary vector is in $\underbrace{\text{linear time}}_{\text{using SMAWK algorithm}}$

Corollary: $O(n+st)$ time suffices to solve Knapsack for all capacities $\leqslant t$

PUTTING PIECES TOGETHER

1. Compute maximal prefix solution P — $O(n)$
2. Solve Knapsack for complement of P, for all capacities $\leqslant t' = 2s^2$ — $O(st') = O(s^3)$
3. Solve $\underbrace{\text{“Negative” Knapsack}}_{\text{find minimum total value of items with total size} \geqslant t}$ for P, for all capacities $\leqslant t' = 2s^2$ — $O(st') = O(s^3)$
4. Combine 2 and 3 — $O(t') = O(s^2)$

OPEN PROBLEM

Close the gap between s^2 and s^3

SUBSET SUM PROBLEM

Given:
- n integers $s_1, s_2, \ldots, s_n$
- target value t

Find: a subset of integers that sum up to exactly t

(Can be reduced to Knapsack by setting $v_i = s_i$)

PSEUDOPOLYNOMIAL TIME ALGORITHMS

$O(nt)$ [Bellmann, 1957]

$\tilde{O}(n+t)$ [Bringmann, 2017]

$O(\mathrm{poly}(n)\, t^{1-\varepsilon})$ impossible* when $s \in \tilde{\Theta}(t)$ [Cygan et al., 2012]
* unless SETH and Set Cover Conjecture fail [Abboud et al., 2019]

Big open problem: Can we get $\tilde{O}(n+s)$ time?

OUR RESULT

$\tilde{O}(n+s^{5/3})$ time algorithm for Subset Sum

Also works for succinctly encoded multisets (i.e., variant with multiplicities)

WARM-UP: QUADRATIC TIME ALGORITHM

1. Take our $O(n+s^3)$ time Knapsack algorithm.
2. Change the $O(n+st)$ time Knapsack algorithm to a $\tilde{O}(n+t)$ time Subset Sum algorithm.
3. Get a $\tilde{O}(n+s^2)$ time Subset Sum algorithm.

INGREDIENT #3: ADDITIVE COMBINATORICS

Let $\lambda \in \tilde{\Theta}(\mu s \Sigma / n^2)$, where $\underbrace{\mu = \text{maximum multiplicity}}_{\text{even without succinct binary encoding, e.g., } \mu = 1 \text{ only when all items are distinct}}$, and $\Sigma = s_1 + s_2 + \cdots + s_n$

Theorem: [Bringmann and Wellnitz, 2021]
Subset Sum can be solved in $\tilde{O}(n)$ time for $t \in (\lambda, \Sigma - \lambda)$

Corollary: Subset Sum can be solved in $\tilde{O}(n + s^{3/2}\mu^{1/2})$ time

Proof of Corollary: use Theorem, falling back to $\tilde{O}(t)$ algorithm if $t \leqslant \lambda$ (w.l.o.g. $t \leqslant \Sigma/2$)

PUTTING PIECES TOGETHER

1. Compute maximal prefix solution P — $O(n)$
2. Solve Subset Sum for complement of P, for all targets $\leqslant t' = 2s^2$, as follows
   (a) Group identical items into *bags* of exactly $s^{1/3}$ items; leave $O(s^{1/3})$ *spares* of each
   (b) Create two (sub-)instances of Subset Sum: *bags instance* and *spares instance*
   (c) In *bags instance*
   - divide everything by $s^{1/3}$;
   - solve for all targets $\leqslant t'' = t'/s^{1/3} = 2s^{5/3}$ — $\tilde{O}(t'') = \tilde{O}(s^{5/3})$

   (d) In *spares instance*
   - $\mu \leqslant O(s^{1/3})$;
   - solve using additive combinatorics — $\tilde{O}(s^{3/2}\mu^{1/2}) = \tilde{O}(s^{5/3})$
3. Solve Subset Sum for P, for all targets $\leqslant t' = 2s^2$, as above — $\tilde{O}(s^{5/3})$
4. $\underbrace{\text{Combine 2 and 3}}_{\text{(not as simple as for Knapsack)}}$ — $O(s^{5/3})$

OPEN PROBLEM

Close the gap between s and $s^{5/3}$

EPFL · Maastricht University · Universität des Saarlandes · mpi max planck institut informatik