KNAPSACK:

Given: A set of n objects with nonnegative integral weights $w_1, \ldots, w_n$ and profits $p_1, \ldots, p_n$, and a capacity $B \geq 0$ of the knapsack.

Find: A subset that respects the weight restriction and maximizes the profit, i.e., a subset $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} w_i \leq B$ and

$$\sum_{i \in I} p_i = \max\{\sum_{i \in I'} p_i \mid I' \subseteq \{1, \ldots, n\} \text{ and } \sum_{i \in I'} w_i \leq B\}.$$

Algorithm (KNAPSACK PACKING)

`Input:` weights $w_1, \ldots, w_n$, profits $p_1, \ldots, p_n$, capacity B

`Output:` $\max\{p^t x \mid w^t x \leq B, x \in \{0,1\}^n\}$.

{ *Initialization* }

$p := \sum_{i=1}^{n} p_i$;

for $t := 1$ **to** p_1 **do** $f[1,t] := w_1$;

for $t := p_1 + 1$ **to** p **do** $f[1,t] := \infty$;

{ *Recursion* }

for $i := 2$ **to** n **do**

  for $t := 1$ **to** p **do**

    if $t \leq p_i$ **then**

      $f[i,t] := \min\{f[i-1,t], w_i\}$

    else

      $f[i,t] := \min\{f[i-1,t], f[i-1,t-p_i] + w_i\}$;

{ *Determine optimal value* }

return $\max\{t \mid f[n,t] \leq B\}$.

The input to a scheduling problem consists of n jobs $J_1, \ldots, J_n$.

The jobs come with processing times $p_1, \ldots, p_n \in \mathbb{N}$.

Each job has to be assigned (scheduled) to one of m identical machines $M_1, \ldots, M_m$ in such a way that the maximum load of a machine is minimized.

The *load* of a machine is the sum of the processing times of all jobs assigned to that machine.

If we assume that each machine can only work on one job at a given time (and will never be idle if there is a job which is not yet processed) then the maximum load corresponds to the maximum completion time of all jobs. This value is also called the *makespan* of the schedule.

Machine Scheduling:

Given: A set of n jobs with positive integral processing times $p_1, \ldots, p_n$, and a number of machines m.

Find: A schedule assigning each of the n jobs to exactly one of the m machines minimizing the makespan.

The New York Times

April 7, 1992

New Shortcut Found For Long Math Proofs

In a discovery that overturns centuries of mathematical tradition, a group of graduate students and young researchers has discovered a way to check even the longest and most complicated proof by scrutinizing it in just a few spots [...]

Using this new result, the researchers have already made a landmark discovery in computer science. They showed that it is impossible to compute even approximate solutions for a large group of practical problems that have long foiled researchers [...]

Theorem. There exists $\varepsilon > 0$ and two functions f_r and g_r, computable in polynomial time, such that for every instance $I \in \text{SAT}$ the following properties are fulfilled:

(1) $f_r(I)$ is an instance of 3-SAT and has $m :=$ poly $(|I|)$ clauses.

(2) If I is satisfiable, then there exists a truth assignment for $f_r(I)$.

(3) If I is not satisfiable, then every truth assignment satisfies at most $(1-\varepsilon)m$ many clauses of $f_r(I)$.

(4) Given a truth assignment τ which satisfies more than $(1-\varepsilon)m$ clauses of $f_r(I)$, $g_r(\tau)$ is a satisfying truth assignment of I.

Corollary. $3\text{-SAT} \notin \mathcal{PAS}$.