Optimierungsprobleme

Ein Optimierungsproblem Π kann man als ein 4-Tuple $\langle \mathcal{I}, Sol, val, \text{goal} \rangle$ beschreiben:

- $\mathcal{I} \subseteq \Sigma^*$ ist die Menge der *Instanzen*; wobei wir annehmen, dass man die Frage „Ist $x \in \mathcal{I}$?“ in polynomieller Zeit entscheiden kann.
- Zu jeder Instanz $I \in \mathcal{I}$ bezeichnet $Sol(I) \subseteq \Sigma^*$ die (nichtleere) Menge der zulässigen *Lösungen*.
- Zu jeder Instanz I und Lösung $x \in Sol(I)$ ist $val(I, x)$ eine positive ganze Zahl. Die Funktion $val(\cdot, \cdot)$ nennt man *Zielfunktion*.
- goal $\in \{\min, \max\}$.

Optimierungsprobleme – Beispiele

- $\textsc{MaxSat}$:

  Gegeben: Boolesche Formel F in konjunktiver Normalform.

  Gesucht: Belegung, die möglichst viele Klauseln erfüllt.

  Hier setzt man daher:

$$
\begin{aligned}
\mathcal{I} &:= \text{Menge aller Boolesche Formeln in konjunktiver Normalform,} \\
Sol(I) &:= \text{Menge der (nicht unbedingt erfüllenden) Belegungen aller Variablen in } I\,, \\
val(I,x) &:= \text{Anzahl Klauseln } I \text{ die von der Belegung } x \text{ erfüllt werden,} \\
\text{goal} &:= \max\,.
\end{aligned}
$$

- $\textsc{MinTsp}$:

  Gegeben: (Vollständiger) Graph $G = (V; E)$, Gewichtsfunktion $w : E \to \mathbb{N}$.

  Gesucht: Hamiltonkreis H mit $\sum_{e \in H} w(e) \stackrel{!}{=} \min$.

  Hier gilt:

$$\begin{aligned} \mathcal{I} &:= \text{Menge aller (vollständigen) gewichteten Graphen,} \\ Sol(G) &:= \text{Menge aller Hamiltonkreise in } G, \\ val(G, H) &:= \text{Länge des Hamiltonkreises } H \text{ in } G, \\ \text{goal} &:= \min . \end{aligned}$$

- **Knapsack Packing:**

  Gegeben: Gewichte $w_1, \ldots, w_n$, Profite $p_1, \ldots, p_n$, Kapazität B.

  Gesucht: $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} p_i = \max\{p^t x \mid w^t x \leq B, x \in \{0,1\}^n\}$ und
  $$\sum_{i \in I} w_i \leq B.$$

  Hier gilt:

$$
\begin{aligned}
\mathcal{I} &:= \text{Menge aller Tupel } (p_1, \ldots, p_n, w_1, \ldots, w_n, B), \\
Sol((...)) &:= \text{Menge aller Teilmengen } I \subseteq \{1, \ldots, n\} \text{ mit } \textstyle\sum_{i \in I} w_i \leq B, \\
val((...), I) &:= \textstyle\sum_{i \in I} p_i, \\
\text{goal} &:= \max.
\end{aligned}
$$

Knapsack Packing

Eingabe: Gewichte $w_1, \ldots, w_n$,
Profite $p_1, \ldots, p_n$, Kapazität B

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

// Initialisierung

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

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

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

// Rekursion

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

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

// Bestimme optimalen Wert

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

Definition: Sei $\Pi = \langle \mathcal{I}, Sol, val, \text{goal} \rangle$ ein Optimierungsproblem. Ein Approximationsalgorithmus *mit absolutem Fehler* B ist ein Approximationsalgorithmus $\mathcal{A}$, für den gilt

$$|\mathcal{A}(I) - \textsc{Opt}(I)| \;\leq\; B \qquad \text{für alle } I \in \mathcal{I}.$$

Definition: Sei $\Pi = \langle \mathcal{I}, Sol, val, \text{goal} \rangle$ ein Optimierungsproblem. Ein Approximationsalgorithmus *mit relativem Fehler* ρ ist ein Approximationsalgorithmus $\mathcal{A}$, für den gilt

$$\frac{|\mathcal{A}(I) - \textsc{Opt}(I)|}{\textsc{Opt}(I)} \;\leq\; \rho \qquad \text{für alle } I \in \mathcal{I}.$$

Definition: Sei $\Pi = \langle \mathcal{I}, Sol, val, \text{goal} \rangle$ ein Optimierungsproblem. Ein Approximationsalgorithmus *mit (relativer) Güte* ρ ist ein Approximationsalgorithmus $\mathcal{A}$, für den gilt

$$\frac{1}{\rho} \;\leq\; \frac{\mathcal{A}(I)}{\textsc{Opt}(I)} \;\leq\; \rho \qquad \text{für alle } I \in \mathcal{I}.$$

Definition: Ein Algorithmus $\mathcal{A}$ heißt *polynomielles Approximationsschema* für ein Optimierungsproblem $\Pi = \langle \mathcal{I}, Sol, val, \text{goal} \rangle$, falls $\mathcal{A}$ für jede Instanz $I \in \mathcal{I}$ and jedes rationale $\epsilon > 0$ eine Lösung $x_{\mathcal{A}} \in Sol(I)$ ausgibt, so dass für $\mathcal{A}(I, \epsilon) := val(I, x_{\mathcal{A}})$ gilt:

$$\frac{1}{1+\epsilon} \leq \frac{\mathcal{A}(I, \epsilon)}{\textsc{Opt}(I)} \leq 1 + \epsilon.$$

Für jedes feste $\epsilon > 0$ muss die Laufzeit von $\mathcal{A}$ polynomiell in $|I|$ sein.

Ist die Laufzeit sogar polynomiell in $|I|$ *und* $1/\epsilon$, so nennt man $\mathcal{A}$ ein *vollständig polynomielles Approximationsschema*.

Definition:

$\mathcal{PO}$ bezeichnet die Menge derjenigen Optimierungsprobleme in $\mathcal{NPO}$, die in polynomieller Zeit optimal gelöst werden können.

$\mathcal{FPTAS}$ bezeichnet die Menge derjenigen Optimierungsprobleme in $\mathcal{NPO}$, für die es ein vollständig polynomielles Approximationsschema gibt.

$\mathcal{PTAS}$ bezeichnet die Menge derjenigen Optimierungsprobleme in $\mathcal{NPO}$, für die es ein polynomielles Approximationsschema gibt.

$\mathcal{APX}$ bezeichnet die Menge derjenigen Optimierungsprobleme in $\mathcal{NPO}$, für die es einen Approximationsalgorithmus mit Güte ρ für *irgendein* $\rho \geq 1$ gibt.

Klar:

$$\mathcal{PO} \subseteq \mathcal{FPTAS} \subseteq \mathcal{PTAS} \subseteq \mathcal{APX} \subseteq \mathcal{NPO}.$$