Worst-Case Analyse von Heuristiken

Ausgang: Minimierungsproblem, Heuristik

Gesucht ist eine *obere Schranke* für den Quotienten aus dem Zielfunktionswert der heuristischen Lösung und dem optimalen Zielfunktionswert in Abhängigkeit von der Problemgröße, die für jede Probleminstanz gültig ist (*pessimistische* oder *Worst-Case* Abschätzung!).

Satz 1. *Für die LPT-Regel angewandt auf Problem $P \mid\mid C_{max}$ gilt:*

$$\frac{C_{max}(LPT)}{C_{max}(OPT)} \leq \frac{4}{3} - \frac{1}{3m}.$$

Frage: Gibt es eine Probleminstanz, für die der Worst-Case eintritt?

Beispiel 1. Für folgende Probleminstanz des Problems $P \mid\mid C_{max}$ tritt der Worst-Case bei Anwendung der LPT-Regel ein: $m = 3$ und

i	1	2	3	4	5	6	7
p_i	5	5	4	4	3	3	3

Approximationsschemata

Ausgang: Minimierungsproblem

Definition 1: Eine Heuristik heißt *ε-Approximationsschema*, falls die Heuristik zu jeder Probleminstanz und jedem $\varepsilon > 0$ eine Lösung x^{HEU} liefert, die

$$\frac{f(x^{HEU})}{f(x^{OPT})} \leq 1 + \varepsilon \qquad \text{d.h.} \qquad \frac{f(x^{HEU}) - f(x^{OPT})}{f(x^{OPT})} \leq \varepsilon \qquad \text{erfüllt.}$$

Definition 2: Eine Heuristik heißt *polynomiales Approximationsschema (PTAS)*, falls ihre Laufzeit für ein festes ε durch ein Polynom der Inputlänge des Problems beschränkt ist.

Definition 3: Ein polynomiales Approximationsschema heißt *vollpolynomiales Approximationsschema (FPTAS)*, falls darüber hinaus die Laufzeit durch ein Polynom in $\frac{1}{\varepsilon}$ beschränkt ist.

Aufwandsvergleich von Approximationsschemata

Beispiel 2. Die folgenden Komplexitäten sollen die Unterschiede zwischen polynomialen und vollpolynomialen Approximationsschemata verdeutlichen:

$$\left.\begin{array}{l} O(\frac{1}{\varepsilon} \cdot n^4) \\ O(n^{\frac{1}{\varepsilon^2}}) \\ O(n^{2 \cdot \frac{1}{\varepsilon}}) \end{array}\right\} \begin{array}{l} \text{alle polynomial in } n, \\ \text{mit } \varepsilon = 0,1 \text{ folgt daraus} \end{array} \left\{\begin{array}{l} O(10 \cdot n^4) \\ O(n^{100}) \\ O(n^{1024}) \end{array}\right.$$

Nur der Ausdruck $O(\frac{1}{\varepsilon} \cdot n^4)$ ist durch ein Polynom in n und $\frac{1}{\varepsilon}$ beschränkt.

Vollpolynomiale Approximationsschemata sind die *besten* Lösungsverfahren, die man für ein Problem der Klasse $\mathbb{NP}$-hard erwarten kann. Leider ist die Existenz solcher Verfahren unter $\mathbb{P} \neq \mathbb{NP}$ nicht für jedes Problem gesichert.

Entwicklung von vollpolynomialen Approximationsschemata

Gegeben: Pseudopolynomialer Algorithmus für ein Problem der Klasse $\mathbb{NP}$-hard

Strategien zur Entwicklung von vollpolynomialen Approximationsschemata:

1. *Runden der Inputdaten*;
2. *Intervallteilung*;
3. *Separierung*.

Beispiel 3. Betrachtet wird das folgende binäre Optimierungsproblem:

$$f(x) = \sum_{i=1}^{n} c_i x_i = \max! \text{ unter } \sum_{i=1}^{n} a_{ij} x_i \leq b_j, j = 1, \ldots, m;\ x_i \in \{0, 1\}, i = 1, \ldots, n.$$

Vorausgesetzt wird, dass für alle i, j gilt: $c_i \geq 0$, $a_{ij} \geq 0$ und $a_{ij} < b_j$.

Enumerative Lösung des Problems

Schrittweise Enumeration: Lege im Schritt k den Wert der Variablen x_k fest ($O(2^k)$)!

Es entstehen bis zum Schritt k Teillösungen der Form $x_i = y_i,\ 1 \leq i \leq k$.

Betrachtet werden zwei verschiedene Teillösungen:
$x_i = y_i,\ 1 \leq i \leq k$ *(Teillösung 1)* und $x_i = z_i,\ 1 \leq i \leq k$ *(Teillösung 2)*,
wobei $\sum_{i=1}^{k} c_i y_i = \sum_{i=1}^{k} c_i z_i$ und $y_i \neq z_i$ für mindestens ein $i \in \{1, \ldots, k\}$ gilt.

Dominanzkriterium: Angenommen, es existiert eine vollständige Lösung $x_i = y_i$, $1 \leq i \leq n$, die Teillösung 1 enthält, so dass für jede vollständige Lösung $x_i = z_i$, $1 \leq i \leq n$, die Teillösung 2 enthält, die Ungleichung

$$\sum_{i=1}^{n} c_i y_i \geq \sum_{i=1}^{n} c_i z_i$$

erfüllt ist. Dann dominiert Teillösung 1 über Teillösung 2 (welche eliminiert werden kann).

Äquivalentes Scheduling-Problem (1)

Die Aufträge $A_i,\ i = 1, \ldots, n$, sollen auf einer Maschine bearbeitet werden. Gegeben sind die Bearbeitungszeiten p_i, die Due Dates d_i und der Gewinn g_i, der immer dann erzielt wird, wenn $C_i \leq d_i$ gilt (ansonsten wird kein Gewinn erzielt). O.B.d.A. sei $d_1 \leq d_2 \leq \ldots \leq d_n$.

Gesucht ist eine Reihenfolge der Aufträge (Indizes) mit maximalem Gewinn:

$$f(x) = \textstyle\sum_{i=1}^{n} g_i x_i = \max!$$

unter

$$\textstyle\sum_{i=1}^{k} p_i x_i \leq d_k, k = 1, \ldots, n$$

wobei $x_i = 1$, falls der Auftrag A_i den Due Date einhält, und 0 sonst.

Es gilt:

$$(f(x) = \sum_{i=1}^{n} g_i(1 - U_i) = \sum_{i=1}^{n} g_i - \sum_{i=1}^{n} g_i U_i \to \max!) \leftrightarrow (\tilde{f}(x) = \sum_{i=1}^{n} g_i U_i \to \min!).$$

Äquivalentes Scheduling-Problem (2)

Also: Lösung eines $1 \mid\mid \sum w_i U_i$ Problems

Bekannte Eigenschaft: Es existiert eine optimale Reihenfolge der Aufträge $\pi = (\pi_1, \pi_2)$, wobei π_1 eine Permutation aller pünktlichen Aufträge ist mit

$$\pi_1 = (i_1, i_2, \ldots, i_r) \quad \Longleftrightarrow \quad d_{i_1} \leq d_{i_2} \leq \ldots \leq d_{i_r}.$$

π_2 ist eine Permutation der verspäteten Aufträge in beliebiger Reihenfolge.

Repräsentation von Teillösung $x_i = y_i, 1 \leq i \leq k$, durch ein Tupel $(f; t)$ mit

$$f = \sum_{i=1}^{k} g_i y_i \qquad \text{und} \qquad t = \sum_{i=1}^{k} t_i y_i$$

Bezeichnung: $S^{(k)}$ ist die Menge aller im k-ten Schritt erzeugten Tupel $(f; t)$

Enumerationsalgorithmus für das Beispiel

Enumerationsalgorithmus für das Beispielproblem

Eingabe: $p_i,\ d_i,\ g_i$ für alle $i \in I$, Aufträge sind nach EDD-Regel nummeriert;
Ausgabe: Optimaler Zielfunktionswert f^{opt}; optimale Reihenfolge π.
BEGIN $S^{(0)} = \{(0,0)\}$; **FOR** $k := 1$ **TO** n **DO**
BEGIN
Bilde $S^{(k)}$ aus $S^{(k-1)}$ durch Hinzunahme von A_k, d.h.
$S^{(k)} := S^{(k-1)} \cup \{(f + g_k; t + p_k) \mid (f;t) \in S^{(k-1)}$ und $t + p_k \leq d_k\}$;
streiche in $S^{(k)}$ alle Tupel $(f;t)$, für die ein Tupel $(f';t')$ existiert, das über $(f;t)$ dominiert, d.h. $f' \geq f$ und $t' \leq t$;
END;
bestimme $f^{opt} := \max\{f \mid (f;t) \in S^{(n)}\}$;
bestimme die zu f^{opt} gehörige Lösung $\pi = (\pi_1, \pi_2)$ durch Rückwärtsrechnung (analog zu dynamischer Optimierung), wobei π_1 eine EDD-Reihenfolge der pünktlichen Aufträge und π_2 eine beliebige Reihenfolge der verspäteten Aufträge ist.
END.

FPTAS durch Intervallteilung für das Beispiel (1)

Für die Stufe k des Enumerationsalgorithmus wird definiert:

$$w_k = \max_{S^{(k)}} \left\{ \sum_{i=1}^{k} g_i x_i \right\}.$$

maximaler Gewinn bzgl. $A_1, A_2, \ldots, A_k$

Dann wird das Intervall $[0, w_k]$ in Teilintervalle der Länge

$$\frac{w_k \cdot \varepsilon}{n}$$

eingeteilt, wobei das letzte kleiner sein kann. Für alle Teillösungen, deren f-Wert im gleichen Intervall liegt, wird das *Dominanzkriterium* angewendet (wobei allen Teillösungen eines Intervalls der gleiche f-Wert zugeordnet wird), so dass nur eine Lösung (die mit kleinstem t-Wert) pro Intervall übrig bleibt, dies liefert die Mengen $R^{(k)} \subseteq S^{(k)}$.

Da die Anzahl der Teilintervalle für Stufe k höchstens gleich $\lceil \frac{n}{\varepsilon} \rceil + 1$ ist, gilt:

$$|R^{(k)}| \leq \lceil \frac{n}{\varepsilon} \rceil + 1 \to \sum_{k=1}^{n} |R^{(k)}| = O\left(\frac{n^2}{\varepsilon}\right),$$

d.h. der Algorithmus ist polynomial in n und $\frac{1}{\varepsilon}$.

Der Fehler jeder Zuordnung ist kleiner als die Intervallänge $\frac{w_k \cdot \varepsilon}{n}$, aber additiv über alle Stufen, d.h.

$$f(x^{HEU}) - f(x^{OPT}) \leq \sum_{k=1}^{n} \frac{w_k \cdot \varepsilon}{n} = \frac{\varepsilon}{n} \sum_{k=1}^{n} w_k.$$

Da aber $w_k \leq f(x^{OPT})$ für alle k gilt, folgt

$$f(x^{HEU}) - f(x^{OPT}) \leq \frac{\varepsilon}{n} \cdot n \cdot f(x^{OPT}).$$

Damit gilt: $(f(x^{HEU}) - f(x^{OPT}))/(f(x^{OPT}) \leq \varepsilon$, d.h. die Heuristik ist ein FPTAS.

Komplexität der Approximierbarkeit von Problemen

Papadimitriou & Yannakakis (1991):
APX ist die Menge aller Minimierungsprobleme, für die ein polynomialer Approximationsalgorithmus mit konstaner Gütegarantie existiert.
PTAS ist die Menge aller Minimierungsprobleme, für die ein PTAS existiert.
FPTAS ist die Menge aller Minimierungsprobleme, für die ein FTPAS existiert.
NP bzw. **P** ist die Menge aller Minimierungsprobleme, deren zugeordnete Entscheidungsprobleme in $\mathbb{NP}$ bzw. $\mathbb{P}$ liegen.