Approximationsschemata in Ablaufplanung, Graphentheorie und Geometrie

Strukturierung der Eingabe

Jürgen Graf

Sommersemester 2004

Übersicht

1. Einführung
2. Makespan auf zwei Prozessoren
3. Verspätung bei einem Prozessor
4. Schlußwort

Problemstellung

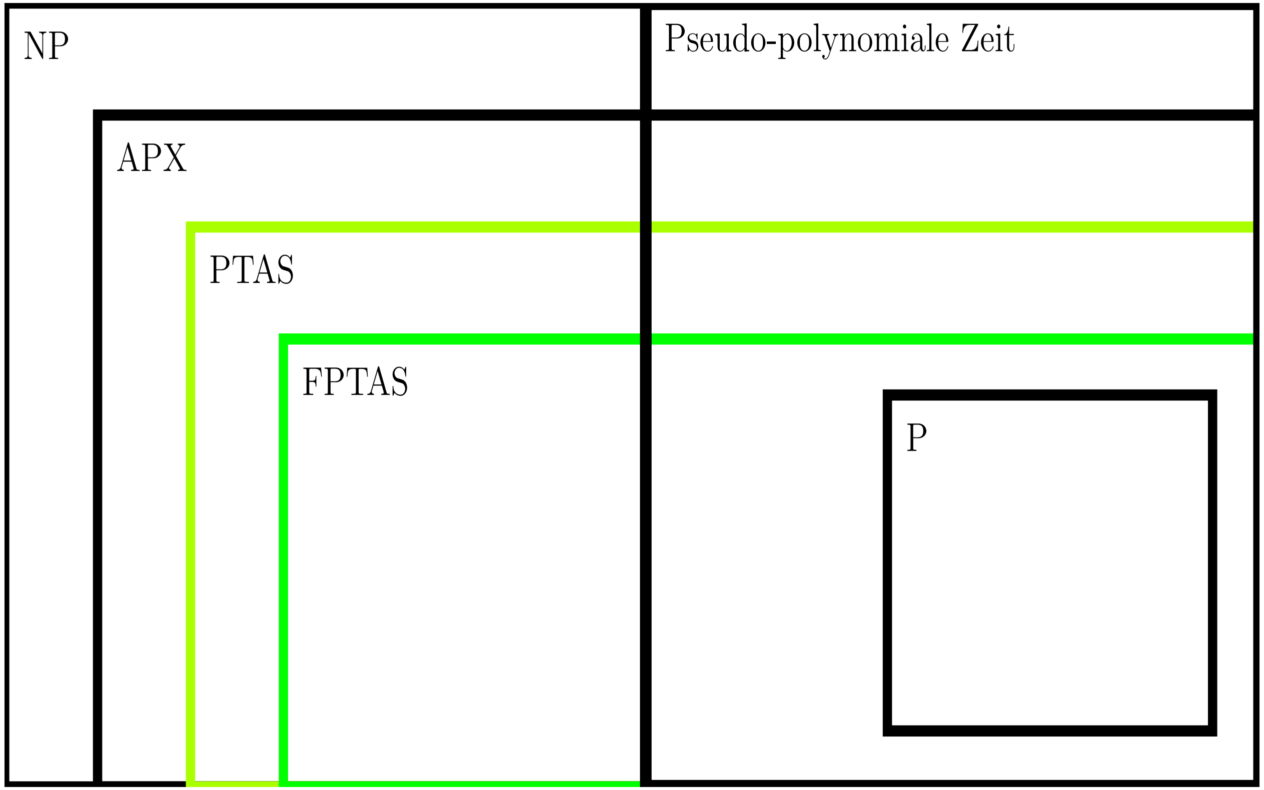
– Optimale Ablaufplanung ist NP-schwer

– Ziel ist ein Approximationsschema

– Methode: Strukturierung der Eingabe

– Einführung weiterer Komplexitätsklassen

Einführung



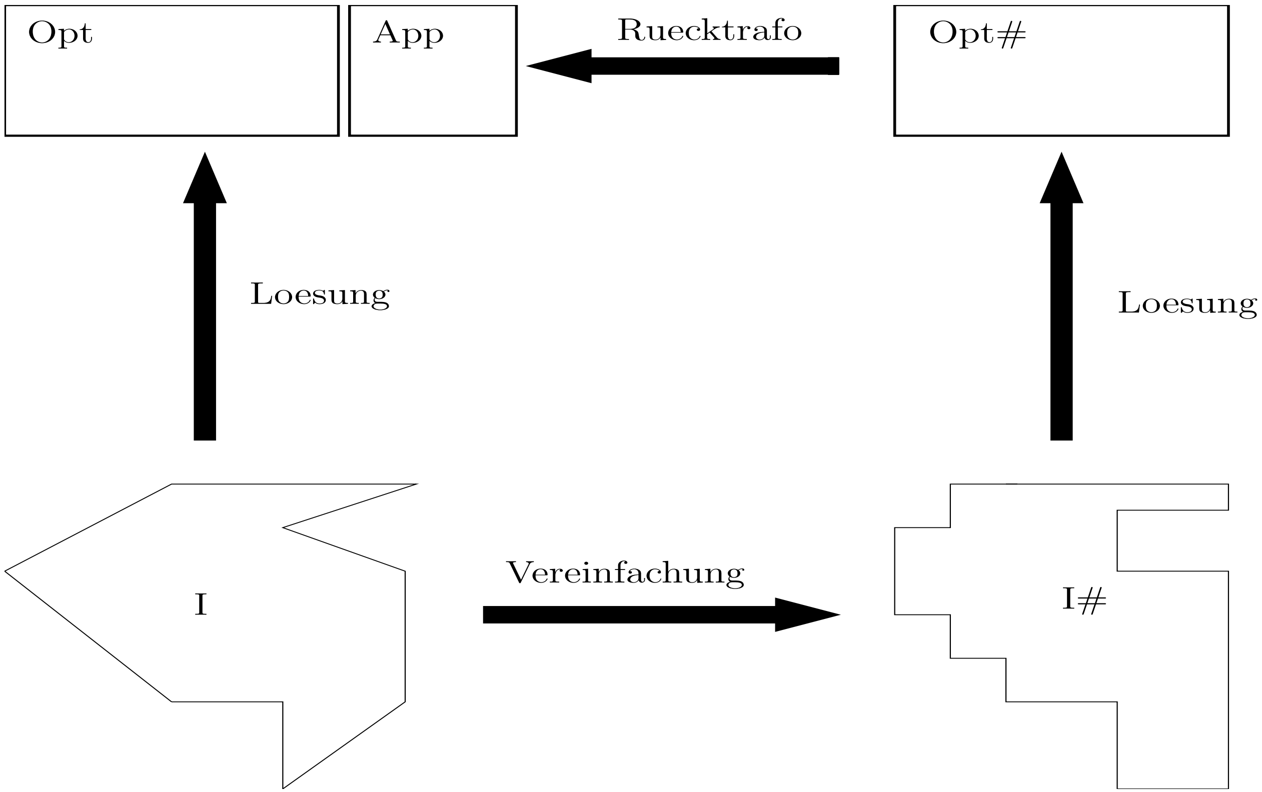
Was kennzeichnet die Strukturierung?

– Strukturierung der Eingabedaten zur besseren Handhabung

– Strukturierung ist ein Balanceakt

– Strukturierung ist $\mathcal{KUNST}$

Wie ist der Ablauf festgelegt?



Übersicht

1. Einführung
2. Makespan auf zwei Prozessoren
3. Verspätung bei einem Prozessor
4. Schlußwort

Makespan auf zwei Prozessoren

Wie wird Strukturierung auf Probleme angewandt?

Voraussetzungen

– Prozessoren sind identisch

– n Eingabeprozesse: $I = \{p_1, p_2, \ldots, p_n\}$

– p_j positive ganzzahlige Verarbeitungszeit

– $p_{\max}$ notiert die Dauer des längsten Prozesses

– Alle Prozesse zum Zeitpunkt Null verfügbar

– Verdrängung nicht erlaubt

Makespan auf zwei Prozessoren

Lemma

Für $L = \max\{\frac{1}{2}p_{\mathrm{sum}}, p_{\max}\}$ gilt $L \leq \mathrm{Opt}$.

Beweisidee

– Gesamtlänge aller Prozesse läßt sich halbieren

– $p_{\max} \leq \mathrm{Opt}$, da längster Prozess verarbeitet werden muß

Vereinfachung

– Übersetzung von I nach $I^{\#}$

– Klassifizierung: $I = G \cup K$

– $0 \leq \epsilon \leq 1$

– $p_j \in \mathrm{G}$ gdw. $p_j > \epsilon L$, sonst $p_j \in \mathrm{K}$.

– $\mathrm{G} \subset I^{\#}$

Makespan auf zwei Prozessoren

Aber wie verfährt man mit den kleinen Prozessen?

Vereinfachung

– Sei Z_K die Gesamtzeit aller kleinen Prozesse

– Setze kleine Prozesse zusammen und teile den entstehenden großen Prozess zu Stücken der Länge ϵL

– $I^{\#}$ besteht aus $\lfloor Z_K/(\epsilon L) \rfloor$ Prozessen der Länge ϵL

Wieso ist $I^{\#}$ eine vereinfachte Instanz von I?

- $\sum$ Zeitabschnitte in $I^{\#} \approx \sum$ Längen der kleinen Prozesse
- Sandkörnermodell
- Man erhält die wichtigsten Eigenschaften der kleinen Prozesse!

Wieso liegt $I^{\#}$ nahe bei I?

Sei $Z_{K,i}$ die Gesamtzeit aller kleinen Prozesse auf einem Prozessor i

Ersetze die kl. Prozesse durch $\lceil Z_{K,i}/(\epsilon L) \rceil$ Intervalle der Länge ϵL

Es gilt:

$$\lceil Z_{K,1}/(\epsilon L) \rceil + \lceil Z_{K,2}/(\epsilon L) \rceil \geq \lfloor Z_{K,1}/(\epsilon L) + Z_{K,2}/(\epsilon L) \rfloor = \lfloor Z_K/(\epsilon L) \rfloor$$

Wieso liegt $I^{\#}$ nahe bei I?

Steigerung der Prozessorbelastung um:

$$\lceil Z_{K,i}/(\epsilon L)\rceil \epsilon L - Z_{K,i} \leq (Z_{K,i}/(\epsilon L) + 1)\epsilon L - Z_{K,i} = \epsilon L$$

Der somit erhaltene Ablaufplan ist gut zu verwalten!

Wieso liegt $I^{\#}$ nahe bei I?

Es gilt:

$$\mathrm{Opt}^{\#} \leq \mathrm{Opt} + \epsilon L \leq (1+\epsilon)\mathrm{Opt}$$

Makespan auf zwei Prozessoren

Wie löst man die vereinfachte Instanz $I^{\#}$?

Lösung

– Für die Verarbeitungszeit gilt: $p_{\mathrm{sum}} \leq 2L$

– Jeder Prozess in $I^{\#}$ hat eine Mindestlänge von ϵL

– Die Anzahl der Prozesse in $I^{\#}$ beträgt $2L/(\epsilon L) = 2/\epsilon$

– ϵ steuert die Anzahl der Prozesse

Lösung

– Probiere *alle* möglichen Ablaufpläne

– Es gibt derer $2^{2/\epsilon}$ viele

– Makespan für einen Ablaufplan läßt sich in $O(2/\epsilon)$ bestimmen

Makespan auf zwei Prozessoren

Wie wird nun die Lösung rückübersetzt?

Rücktransformation der Lösung

– L_i bezeichne die Belastung des Prozessors i

– $L_i = Z_{G,i} + Z_{K,i}$

– $Z^{\#}_{K,1} + Z^{\#}_{K,2} = \epsilon L \cdot \lfloor Z_K / (\epsilon L) \rfloor > Z_K - \epsilon L$

Rücktransformation der Lösung

– Die großen Prozesse behalten ihre Position

– Reserviere eine Länge von $Z^{\#}_{K,1} + 2\epsilon L$ auf Prozessor 1

– Reserviere ein Intervall der Länge $Z^{\#}_{K,2}$ auf Prozessor 2

– Fülle zunächst die Freiräume auf Prozessor 1 mit kleinen Prozessen auf

Rücktransformation der Lösung

– Kleine Prozesse sind durch ϵL beschränkt

– Es gilt: $Z - Z^{\#}_{K,1} - \epsilon L \leq Z^{\#}_{K,2}$

– Damit ist der Ablaufplan vollständig beschrieben

Rücktransformation der Lösung

Für die Prozessorbelastung gilt:

$$
\begin{aligned}
L_i &\leq Z^{\#}_{G,i} + Z^{\#}_{K,i} + 2\epsilon L \\
&= L^{\#}_i + 2\epsilon L \\
&\leq \mathrm{Opt}^{\#} + 2\epsilon \mathrm{Opt} \\
&\leq (1+\epsilon)\mathrm{Opt} + 2\epsilon \mathrm{Opt} \\
&= (1+3\epsilon)\mathrm{Opt}
\end{aligned}
$$

Schlußbemerkung zum Beispiel

– Die Lösung ist um maximal $1 + 3\epsilon$ größer als die opt. Lösung

– Die Konstante ϵ läßt sich beliebig festlegen

– Man erhält ein PTAS

Aber wie bekommt man ein FPTAS?

Übersicht

1. Einführung
2. Makespan auf zwei Prozessoren
3. Verspätung bei einem Prozessor
4. Schlußwort

Voraussetzungen

- n Eingabeprozesse
- p_j positive ganzzahlige Verarbeitungszeit
- d_j ganzzahlige Fälligkeitszeitpunkte
- C_j Fertigstellungszeit für einen Prozess
- T_j Verspätung eines Prozesses
- Alle Prozesse zum Zeitpunkt Null verfügbar
- Verdrängung nicht erlaubt

Was will man erreichen?

– Minimierung von $\sum_{j=1}^{n} T_j$

– Minimiere die Gesamtzeit der Verspätungen für alle Prozesse

– Vermeide Leerlaufzeiten zwischen je zwei Prozessen

Voraussetzungen

- $1 \mid\mid \sum T_j$ läßt sich als dynamisches Programm formulieren (Lawler)
- EDD sei *earliest-due-date*
- T_{EDD} sei die maximale Verspätung im EDD-Ablaufplan
- Lösung des dynamischen Programms in $O(n^5 T_{\mathrm{EDD}})$

Voraussetzungen

– EDD-Ablaufplan wird durch EDD-Regel in $O(n \log n)$ erzeugt

– Prozesse nach aufsteigenden Fälligkeitszeitpunkt geordnet

– $0 < T_{\mathrm{EDD}} \leq \mathrm{Opt}$

– Diese Ordnung minimiert $\max T_j$

Verspätung auf einem Prozessor

Wie gestaltet man die Vereinfachung I?

Vereinfachung

– Strukturierung der Eingabe durch σ gesteuert

– $\sigma := \frac{2\epsilon}{n(n+3)} \cdot T_{\mathrm{EDD}}$

– Für die Prozesszeiten gelte: $p_j^{\#} = \lfloor p_j / \sigma \rfloor$

– Für die Fälligkeitszeitpunkte gelte: $d_j^{\#} = \lceil d_j / \sigma \rceil$

Verspätung auf einem Prozessor

Wie löst man die Vereinfachung $I^{\#}$?

Lösung

– Lawlers Algorithmus zum Lösen des dyn. Programms

– Lawlers Algo. hängt von der Anzahl der Prozesse und von T_{EDD} ab

– Umrechnung der Fälligkeitszeitpunke monoton

– Permutationsabbildung π aus I gilt auch in $I^{\#}$

– $T_{\mathrm{EDD}}^{\#} \leq T_{\mathrm{EDD}}/\sigma = n(n+3)/(2\epsilon) = O(n^2/\epsilon)$

– Aufwand für Lawlers Algorithmus somit $O(n^7/\epsilon)$

Verspätung auf einem Prozessor

Wie sieht die Rücktransformation aus?

Rücktransformation der Lösung

– Prozesse seien in einer optimalen Reihenfolge

– Behalte die Positionen der Prozesse unter der Rücktrafo bei

– Strukturelle Ähnlichkeiten zwischen I und $I^{\#}$

Rücktransformation der Lösung

– Es gilt: $p_j \leq \sigma p_j^{\#}$ und $d_j > \sigma d_j^{\#} - \sigma$

– Es gilt für $j = 1, \ldots, n$:

$$
\begin{aligned}
C_j &= \sum_{i=1}^{j} p_i \\
&\leq \sigma \sum_{i=1}^{j} p_i^{\#} + j\sigma \\
&= \sigma \cdot C_j^{\#} + j\sigma
\end{aligned}
$$

Rücktransformation der Lösung

Man gewinnt nun folgende Aussage:

$$
\begin{aligned}
T_j &= \max\{0, C_j - d_j\} \\
&\leq \max\{0, (\sigma \cdot C_j^{\#} + j\sigma) - (\sigma d_j^{\#} - \sigma)\} \\
&\leq \sigma \cdot T_j^{\#} + (j+1)\sigma
\end{aligned}
$$

Rücktransformation der Lösung

Damit ergibt sich:

$$\sum_{j=1}^{n} T_j \leq \sum_{j=1}^{n} \sigma \cdot T_j^{\#} + \frac{1}{2} n(n+3) \cdot \sigma$$

$$\leq Opt + \epsilon T_{\mathrm{EDD}}$$

$$\leq (1+\epsilon)\mathrm{Opt}$$

Schlußfolgerung

– Für jedes ϵ läßt sich ein Ablaufplan in $O(n^7/\epsilon)$ angeben

– Der Ablaufplan läßt sich durch $(1+\epsilon)\mathrm{Opt}$ nach oben abschätzen

– Man erhält somit ein FPTAS

Übersicht

1. Einführung

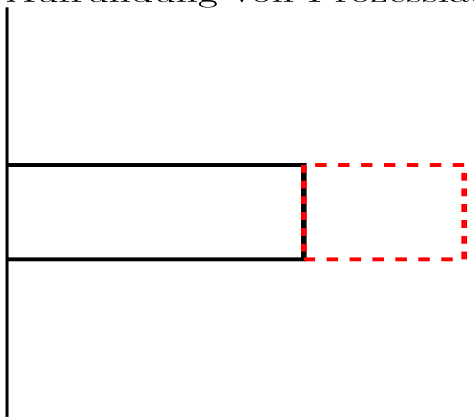
2. Makespan auf zwei Prozessoren

3. Verspätung bei einem Prozessor

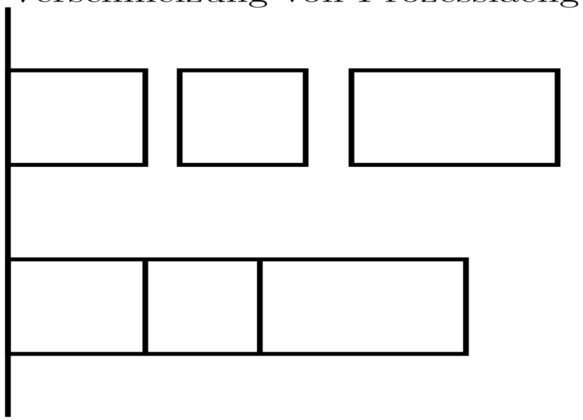
4. Schlußwort

Wie strukturiert man?

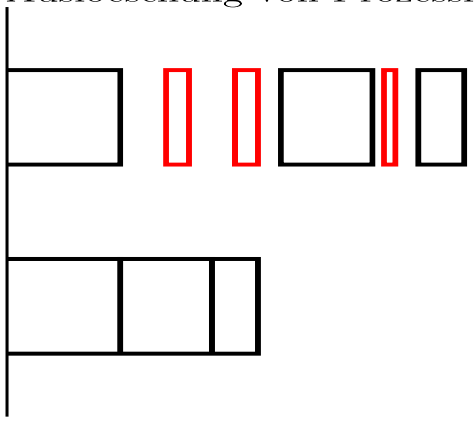
Aufrundung von Prozesslaengen



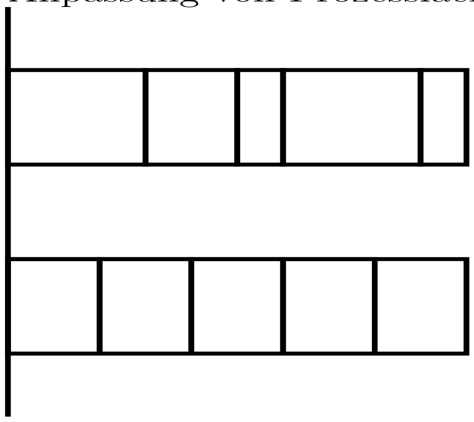
Verschmelzung von Prozesslaengen



Ausloeschung von Prozesslaengen



Anpassung von Prozesslaengen



Rundung von Prozesslängen

– Runde Prozesslängen zu ganzzahligen Zahlen

– Runde Prozesslängen zur nächstliegenden Zweierpotenz

Verschmelzung von Prozesslängen

– Kleinere Einheiten zu größeren Einheit zusammenfassen

– Zusammengefasste Einheit kann angepasst werden

Auslöschung von Prozesslängen

– Störende Längen beseitigen

– Ansammlung kleinerer Prozesse missachten

Anpassung von Prozesslängen

– Betrachte Menge von Prozessen mit ungefähr gleicher Länge

– Finde mittlere Prozesslänge

– Lege entsprechende Kopie von Hülsen dieser Länge an

– Die Hülsen ersetzen die Prozesse

Ende des Vortrags

Vielen Dank für die Aufmerksamkeit!

Noch Fragen ?