Lehrstuhl für Informatik mit
Schwerpunkt Wissenschaftliches Rechnen
Prof. H.-J. Bungartz, M. Buchholz, St. Zimmer

SS 2010

Modellbildung und Simulation
Übungsblatt 3: Scheduling
Lösungsvorschlag

Zu den Übungen am 11.5.2010

1 Ein Fehler in den Folien

Auf Folie 23 wurde vor dem Folienupdate ein kritischer Pfad als ein Pfad von A_S nach A_E definiert, der nur aus kritischen Knoten besteht. Nun ist noch der Zusatz „… und bei dem für jede Kante $A_k \to A_l$ gilt $c'_k = s''_l$ (A_l muss unmittelbar auf A_k folgen) …“ hinzugekommen.

- Zeigen Sie, dass die Behauptung „Alle kritischen Pfade in unserem Graph haben dieselbe Länge“ ohne den Zusatz nicht unbedingt gilt, mit dem Zusatz aber schon.

  Ohne Zusatz können wir in einem Pfadstück $A_1 \to A_2 \to A_3$ eine weitere Kante $A_1 \to A_3$ einfügen, die für das Schedulingproblem irrelevant ist, aber den Pfad um t_2 verkürzt (und wenn er vorher nur kritische Knoten enthalten hat, tut er das hinterher erst recht).

  Mit Zusatz können wir zeigen, dass ein kritischer Pfad ein Pfad maximaler Länge von A_S nach A_E ist, dann haben insbesondere auch alle diese Pfade dieselbe Länge (man kann aber die Gleichheit für verschiedene kritische Pfade auch direkt aus Gleichung (1) ablesen).

  Sei dazu $A_1 \to A_2 \to A_3 \to \ldots \to A_m$ ein kritischer Pfad. Wegen $c'_k = s''_{k+1} \Rightarrow c'_k = s'_{k+1}$ (es ist ja $c'_k \leq s'_{k+1} \leq s''_{k+1}$) und wegen $c'_k = s'_k + t_k$ ist in einem optimalen Zeitplan

$$s_k = \sum_{j=1}^{k-1} t_i. \tag{1}$$

  Insbesondere ist $c_E = s_E = s_m$ die Summe der t_k auf dem kritischen Pfad. Ein längerer Pfad kann die Präzedenzbedingungen unmöglich einhalten (in beliebigen Pfaden gilt ja (1) mit „$\geq$“).

- Zeigen Sie, dass jeder kritische Knoten auf wenigstens einem kritischen Pfad (in der neuen Definition) liegt.

  Für einen kritischen Knoten $A_k \neq A_E$ und einen Nachfolger A_l mit

$$s''_l = \min_{j:A_k \to A_j} s''_j$$

  gilt nach Konstruktion $c''_k = s''_l$ und weil A_k kritisch ist, auch $c'_k = s''_l$. Weiter ist A_l wegen $c'_k \leq s'_l \leq s''_l$ ebenfalls kritisch.

  Fortsetzen dieser Konstruktion bringt uns auf einem kritischen Pfad nach A_E.

  In Richtung A_S geht's analog: sei A_l ein Vorgänger von A_k mit

$$c'_l = \max_{j:A_j \to A_k} c'_j.$$

  Dann ist nach Konstruktion $c'_l = s'_k$ und weil A_k kritisch ist, auch $c'_l = s''_k$. Weiter ist A_l wegen $c'_l \leq c''_l \leq s''_k$ ebenfalls kritisch.

2 Scheduling: Tiefensuche

In einem gerichteten Graphen, bei dem es einen Startknoten S gibt, von dem aus man jeden anderen Knoten A_k erreicht (d.h., es gibt einen Pfad $S \to \ldots \to A_k$), kann man von S aus mittels *Tiefensuche* alle Knoten ablaufen. Jeder Knoten sei dazu färbbar mit Farben Weiß, Grau und Schwarz, am Anfang seien alle Knoten weiß. Dann beginnt man mit einem Aufruf tiefensuche(S) folgender rekursiver Funktion

tiefensuche(A):
1: A grau färben
2: Für alle Nachfolger A' (also die Knoten A' mit $A \to A'$):
3: Wenn A' weiß ist: tiefensuche(A')
4: A schwarz färben

(Ohne ausgezeichneten Startknoten käme noch eine Schleife außenrum, bis alle Knoten bearbeitet sind.)

- Wie muss man den Algorithmus modifizieren, um festzustellen, ob der Graph zyklenfrei ist?

  Vor Schritt 3 nachsehen, ob A' grau ist. Wenn ja, ist man auf einen Zyklus gestoßen (wir sind ja von dem grauen Knoten zu A' gekommen). Tritt das nie ein, ist der Graph zyklenfrei.

- Und wie, um die Reihenfolge einer topologischen Sortierung zu ermitteln?

  Vor Schritt 4 den Knoten A vorne *an eine (anfangs leere) Liste anfügen.*

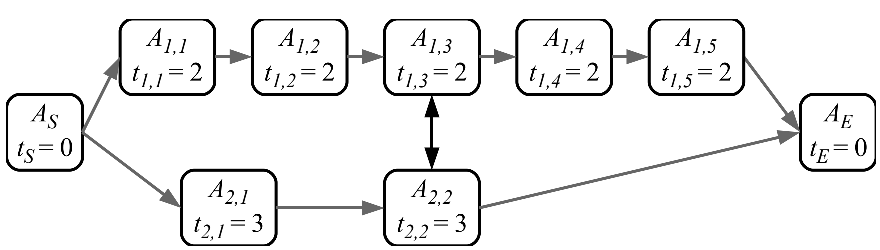
3 Scheduling: job shop

Für die Abwicklung von zwei Aufträgen A_1 und A_2, die als *job shop* abzuarbeiten sind, soll eine optimale Bearbeitungsreihenfolge bestimmt werden.
Ein Auftrag A_j wird beschrieben durch die Bearbeitungszeiten auf den m_j benötigten Maschinen: A_1 besteht aus fünf Teilaufträgen mit einer Bearbeitungszeit von je 2h, A_2 aus zwei Teilaufträgen mit einer Bearbeitungszeit von je 3h.
Zusätzlich benötigen der dritte Teilauftrag $A_{1,3}$ von A_1 und der zweite Teilauftrag $A_{2,2}$ von A_2 dieselbe Maschine.

- Was charakterisiert die Abarbeitung als *job shop*, z.B. im Gegensatz zur Abarbeitung als *open shop* oder *flow shop*?

  Nachlesen: Folie 25

- Modellieren Sie das Schedulingproblem für A_1 und A_2 als Präzendenzgraph.



- Geben Sie für jede der möglichen Disjunktivkantenbelegungen die frühestmöglichen Startzeiten $s'_{i,j}$ der einzelnen Teilaufträge $A_{i,j}$ und des Zielknotens E an.

	$s'_{1,1}$	$s'_{1,2}$	$s'_{1,3}$	$s'_{1,4}$	$s'_{1,5}$	$s'_{2,1}$	$s'_{2,2}$	s'_E
$A_{1,3} \to A_{2,2}$	*0*	*2*	*4*	*6*	*8*	*0*	*6*	*10*
$A_{2,2} \to A_{1,3}$	*0*	*2*	*6*	*8*	*10*	*0*	*3*	*12*