

# Zusammenfassung

## Strukturierung der Eingabe

Jürgen Graf

Die Lösung zur Bestimmung eines optimalen Ablaufplans ist NP-schwer. Um eine effizientere Lösung zu erhalten, liegt die Idee nahe, von dem Anspruch eine optimale Lösung zu erhalten abzuweichen und stattdessen eine approximiertere Lösung zu akzeptieren. Diesen Ansatz zu verfolgen, macht nur dann Sinn, wenn sich eine approximiertere Lösung weitaus schneller generieren läßt als die optimale Lösung des Problems.

In dieser Ausarbeitung soll eine mögliche Herangehensweise aufgezeigt werden. Dabei versucht man durch Strukturierung der Eingabe einen effizienteren Algorithmus zu erhalten. Die Konstruktion eines Algorithmus erfolgt nach einem Schema, welches aus drei Schritten besteht. Der erste Schritt ist die *Vereinfachung* der Eingabe durch *Rundung*, *Verschmelzung*, *Schnitt* oder *Anpassung*. Der zweite Schritt ist die *optimale* Lösung der vereinfachten Eingabe. Der letzte Schritt ist die *Rücktransformation* der gefundenen Lösung. Dabei muss die rücktransformierte Lösung die optimale Lösung des Ablaufplans hinreichend gut approximieren. Läuft alles gut, erhält man auf diesem Weg ein (F)PTAS.

An drei Beispielen aus der Ablaufplanung soll dargestellt werden, wie sich die oben genannten Methoden anwenden lassen. Im ersten Beispiel wird gezeigt, wie man für das Problem  $P2 \parallel C_{\max}$  ein PTAS findet. Bei diesem Problem wird ein minimaler Ablaufplan für eine Maschinenumgebung mit zwei identischen Prozessoren gesucht. Dazu werden die Prozesse zunächst in lange und kurze Prozesse aufgeteilt. Die Entscheidung, ob ein zu verarbeitender Prozess lang oder kurz ist, geschieht mit einer zu Beginn festgelegten Konstante. Mit dieser Konstanten läßt sich der Grad der Strukturierung steuern. Im nächsten Beispiel wird das Problem  $P \parallel C_{\max}$  behandelt. Dieses stellt die Verallgemeinerung des ersten Problems in dem Sinne dar, dass hier  $n$  Prozessoren für die Verarbeitung der Prozesse zur Verfügung stehen. Auch hierfür läßt sich ein PTAS angeben. Dieses hängt jedoch stark von einem beim Algorithmus verwendeten gemischt-ganzzahligen Programm ab. Zuletzt wird am Beispiel des Problems  $1 \parallel \sum T_j$  ein FPTAS vorgestellt. Bei diesem Problem geht es darum die Gesamtzeit der Verspätungen zu minimieren. Wesentliche Schritte zur Erstellung des Algorithmus sind die Induzierung einer Ordnung auf der Menge der Eingabeprozesse durch eine *Fälligkeitsregel* und die Verwendung von *dynamischem Programmieren*.