

Zusammenfassung

Scheduling und *Routing* sind klassische Felder der theoretischen Informatik, welche in den letzten Jahrzehnten große Aufmerksamkeit erhielten. Das Ziel sowohl von Scheduling als auch von Routing ist die effiziente Nutzung von Computersystemen.

Das Ziel von Scheduling ist die Bestimmung einer *optimalen Zuweisung* von *Jobs* auf (einen oder mehrere) *Prozessoren* bezüglich eines Gütemaßes, wohingegen das Ziel von Routing das effiziente Verschicken von (teilbaren oder unteilbaren) *Paketen* durch ein gemeinsames Prozessornetzwerk ist. Die Theorie des Scheduling und Routing ist durch eine nahezu beliebige Anzahl von Problemklassen charakterisiert. In dieser Dissertation analysieren wir drei von ihnen. Da sie unabhängig von einander sind, untersuchen wir sie ausgiebig in drei in sich abgeschlossenen Kapiteln.

- **Scheduling von identischen dehnbaren Jobs:** Wir betrachten das Problem der Berechnung eines Schedules für unabhängige dehnbare *ununterbrechbare* identische Jobs auf identischen Prozessoren mit *minimaler totaler Abschlusszeit*, dem sogenannten *Makespan*.
- **Fluss Scheduling:** Wir betrachten *synchrone verteilte Prozessornetze*. Wir nehmen an, dass die Ladung auf den Prozessoren aus unabhängigen Lasteinheiten, den sogenannten *Tokens*, besteht. Um diese Last im Netzwerk zu balancieren, ist es nötig, Teile der Last einiger Prozessoren zu verteilen. Wir verteilen diese Last mit Hilfe eines gegebenen *balancierenden* Flusses mit dem Ziel, die minimale Anzahl von Runden zu nutzen, um den balancierten Zustand zu erreichen.
- **Eigensinniges Routing in nicht-kooperativen Netzwerken:** Wir betrachten ein Routing-Spiel, bekannt als *KP-Modell*. In diesem Modell verschicken nicht-kooperative *Nutzer* ihre unteilbare *Last* durch ein sehr einfaches Netzwerk, welches nur aus *parallelen Kanten* zwischen 2 *Knoten* besteht. Jeder Nutzer wählt eine gemischte Strategie, um seine *erwartete Latenz* zu minimieren. Ein stabiler Zustand, in dem kein Nutzer einen Anreiz hat seine Strategie einseitig zu ändern, heißt *Nash Equilibrium*. Wir analysieren die Berechenbarkeit von Nash Equilibrien und den Einfluss des eigensinnigen Verhaltens der Nutzer auf das globale System.