

Ausgewählte Kapitel
eingebetteter Systeme

Statische Ablaufplanung

Matthias Bott

Hauptseminar SS 2006

12.07.2006



Übersicht

1. Statische vs. Dynamische Ablaufplanung
2. Strategien zur Erstellung statischer Ablaufpläne
3. Erweiterte statische Ablaufplanung

Statische Ablaufplanung

Alle Tasks und deren Eigenschaften sind vor der Planung bekannt:

- Bereitzeit
- Maximale Ausführungszeit (*WCET*)
- Frist (*Deadline*)
- Periode

Statische Ablaufplanung

Zeitpunkt der Erstellung:

Offline, vor Programmausführung

Ergebnis:

- Vollständiger Ablaufplan in tabellarischer Form
- Wird nicht mehr verändert

Vergleich: Dynamische Ablaufplanung

Menge der Tasks ändert sich zur Laufzeit

→ nur Informationen zu der aktuellen Menge bekannt

Zeitpunkt der Ablaufplanung:

Online, während Programmausführung

Statischer Ablaufplan

Repräsentiert durch Tabelle

Beispiel:

T1	2
T2	4
T3	3
T4	1

Adresse/Identifikation
eines Arbeitsauftrags

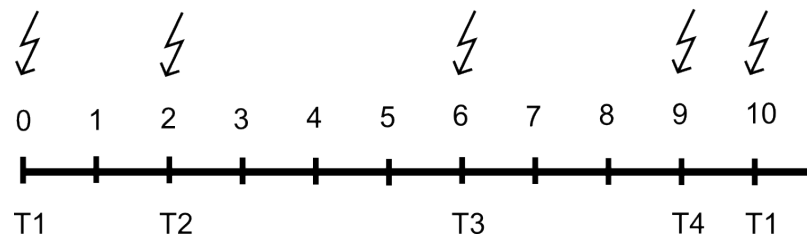
Zeitintervall bis zur nächsten
Planungsentscheidung

Statischer Ablaufplan

Abarbeitung durch Dispatcher

- Er lastet Aufgabe ein
- Stellt Zeitgeber auf Zeitpunkt der nächsten Planungsentscheidung

T1	2
T2	4
T3	3
T4	1

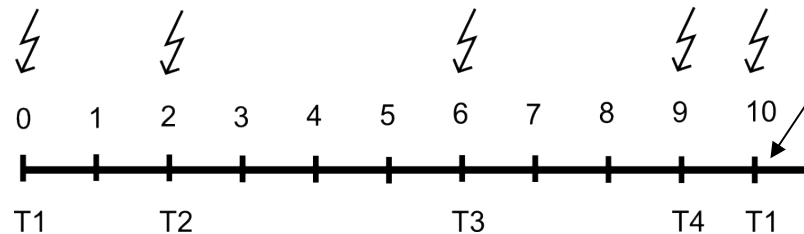


Statischer Ablaufplan

Abarbeitung durch Dispatcher

- Er lastet Aufgabe ein
- Stellt Zeitgeber auf Zeitpunkt der nächsten Planungsentscheidung

T1	2
T2	4
T3	3
T4	1



Ablaufplan wird
zyklisch ausgeführt



Übersicht

1. Statische vs. Dynamische Ablaufplanung
2. Strategien zur Erstellung statischer Ablaufpläne
3. Erweiterte statische Ablaufplanung

Ablaufplanung durch Suche

Ausgegangen wird von:

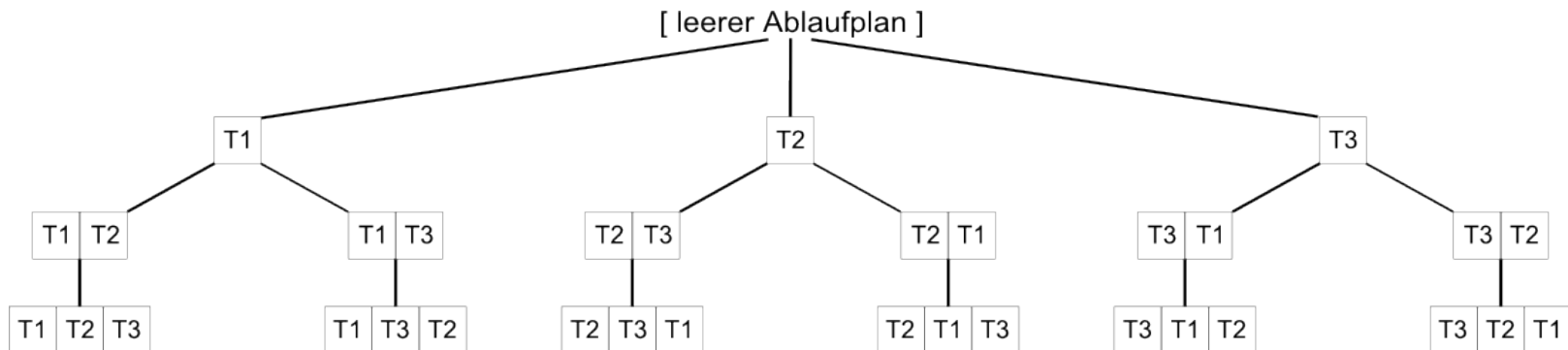
- Periodischen, nicht unterbrechbaren Tasks
- Alle Eigenschaften der Tasks bekannt
(Bereitzeit, Ausführungszeit, Frist,...)

Gesucht:

Statischer Ablaufplan, der alle Bedingungen erfüllt

Erstellung des Suchbaums

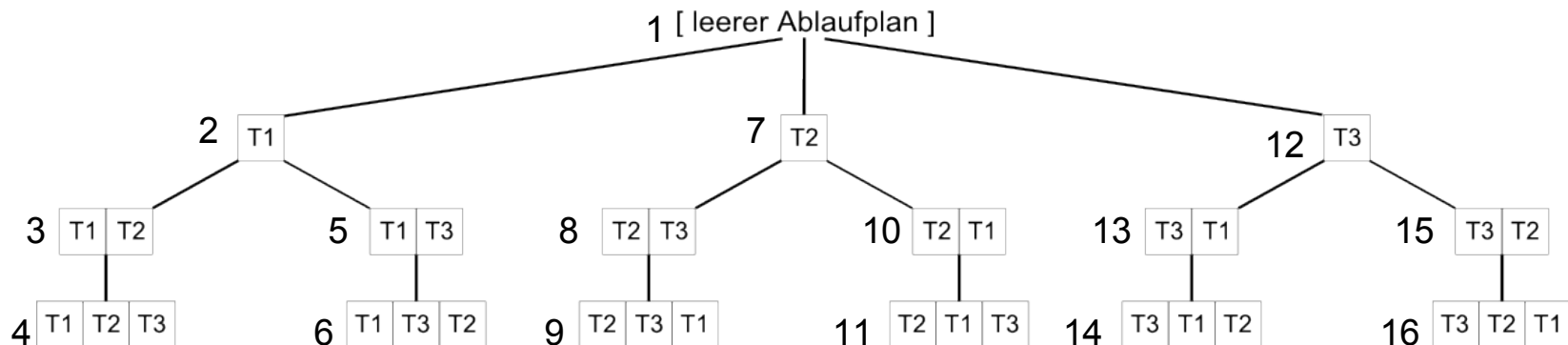
- Wurzel: leerer Ablaufplan
- Blatt: vollständiger Ablaufplan
- Kante: Einplanungsentscheidung eines Tasks
- innerer Knoten: unvollständiger Ablaufplan



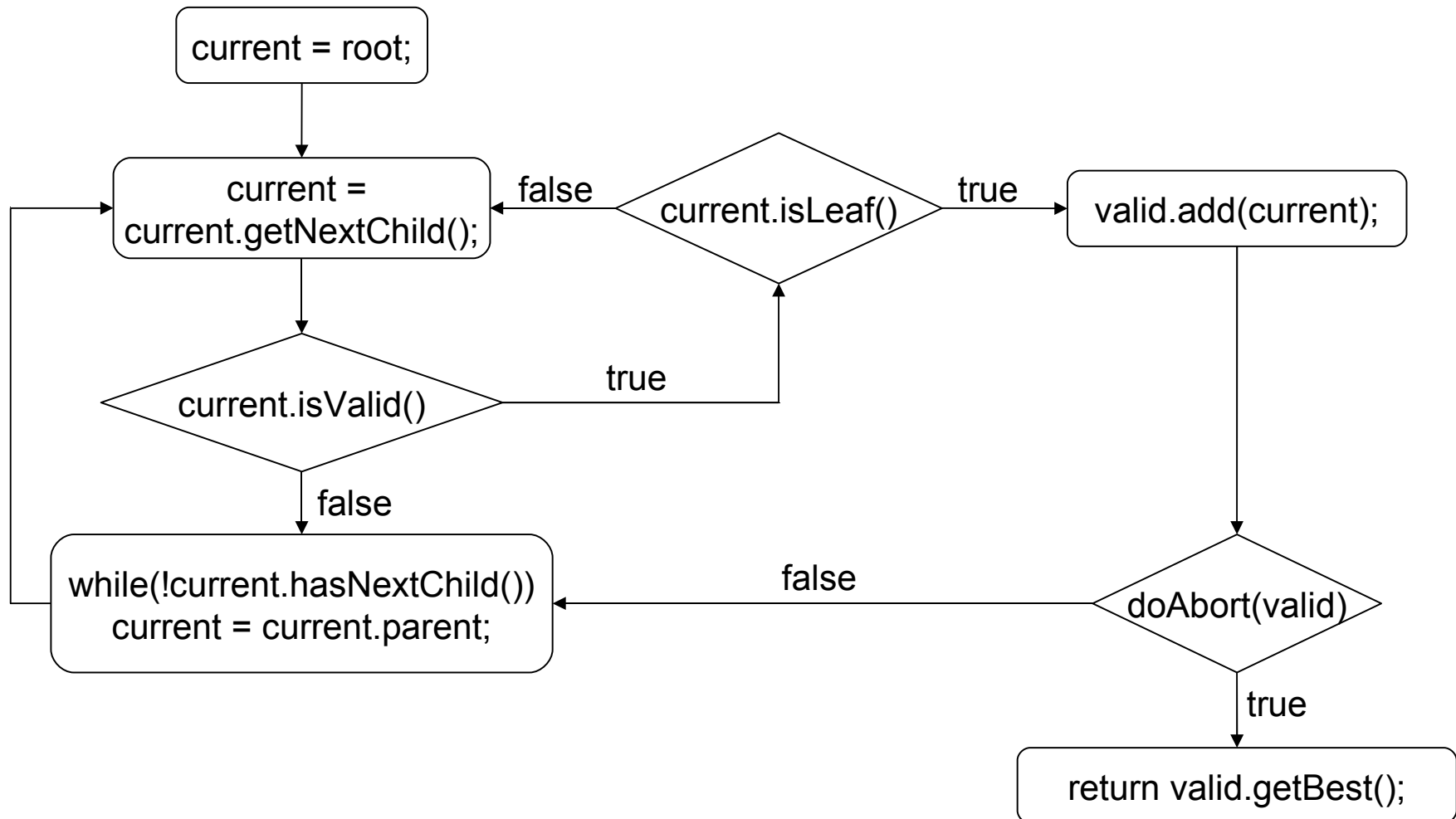
Suche nach gültigen Ablaufplänen

Tiefensuche im Suchbaum

- Jeder (unvollständige) Ablaufplan wird geprüft
- Falls gültig:
Abbruchkriterium entscheidet, ob mit der Suche fortgefahren wird.



Suche nach gültigen Ablaufplänen



Suche nach gültigen Ablaufplänen

Überprüfung der Gültigkeit *current.isValid()*

Werden zeitliche Eigenschaften der Tasks berücksichtigt?

- $\text{Bereitzeit} \leq \text{Startzeit}$
- $\text{Startzeit} + \text{WCET} < \text{Frist}$

Beispiel für Abbruchkriterium *doAbort(valid)*

Kriterium immer erfüllt

→ Abbruch nach erster gefundenen Lösung

Ablaufplanung mit Hilfe der Tabu-Suche

Tabu-Suche ist

- iterativ.
- ein heuristisches Optimierungsverfahren.
- für komplexe Probleme geeignet.

Ablaufplanung mit Hilfe der Tabu-Suche

Gestartet wird mit Initial-Ablaufplan

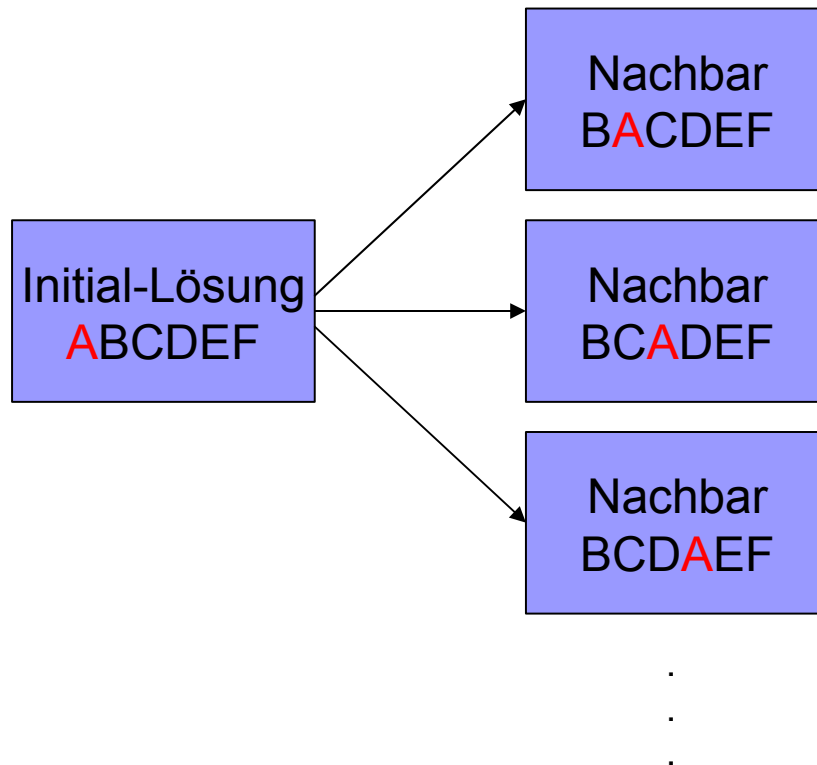
Initial-Lösung
ABCDEF

Tabu-Liste

ABCDEF

Ablaufplanung mit Hilfe der Tabu-Suche

Erzeugung einer „Nachbarschaft“ durch eine Nachbarschaftsfunktion

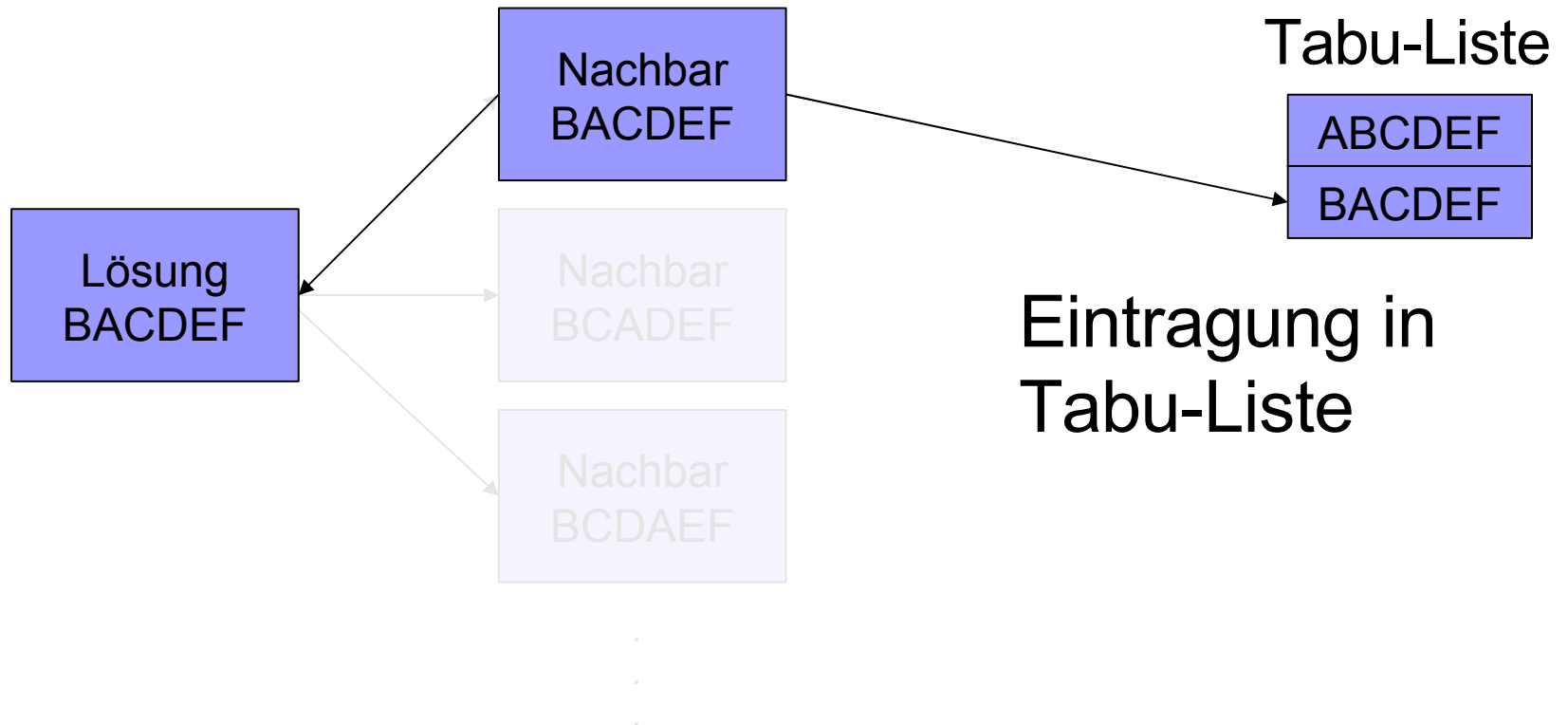


Tabu-Liste

ABCDEF

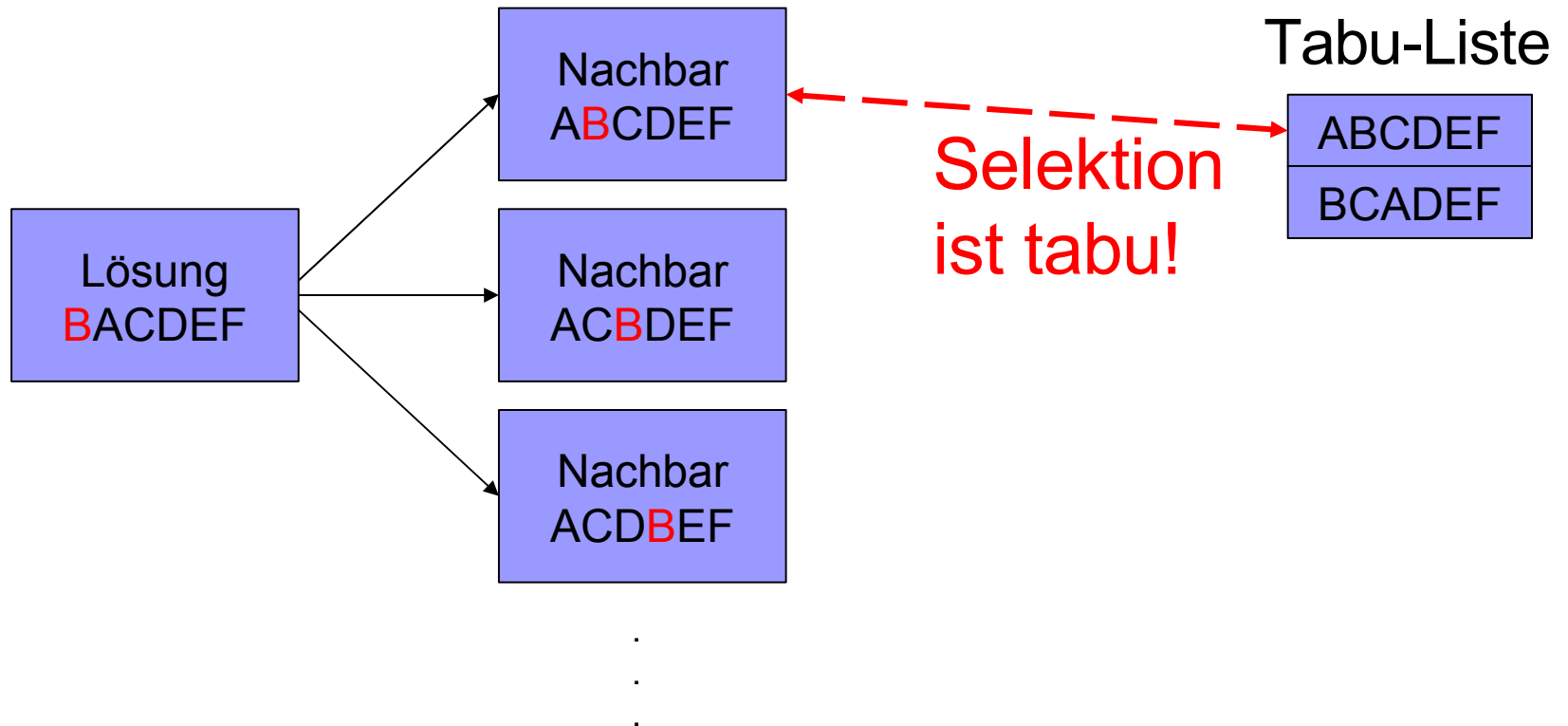
Ablaufplanung mit Hilfe der Tabu-Suche

Auswahl der besten Lösung als Basis für nächste Iteration



Ablaufplanung mit Hilfe der Tabu-Suche

Erzeugung der nächsten Nachbarschaft



Ablaufplanung mit Hilfe der Tabu-Suche

Größe der Tabu-Liste

- Zu klein → Zirkulation zwischen besten Lösungen
- Zu groß → hoher Speicherverbrauch

Ablaufplanung mit Hilfe der Tabu-Suche

Akzeptierung von nicht verbessernden Lösungen
→ Möglichkeit aus lokalen Optima zu entkommen

Bewertungsfunktion

Mögliches Kriterium:

Anzahl eingehaltener Fristen

Weitere heuristische Verfahren

- Simulierte Abkühlung (Simulated annealing)
- Evolutionäre Algorithmen
-



Übersicht

1. Statische vs. Dynamische Ablaufplanung
2. Strategien zur Erstellung statischer Ablaufpläne
3. **Erweiterte statische Ablaufplanung**

Wechsel des Arbeitsmodus

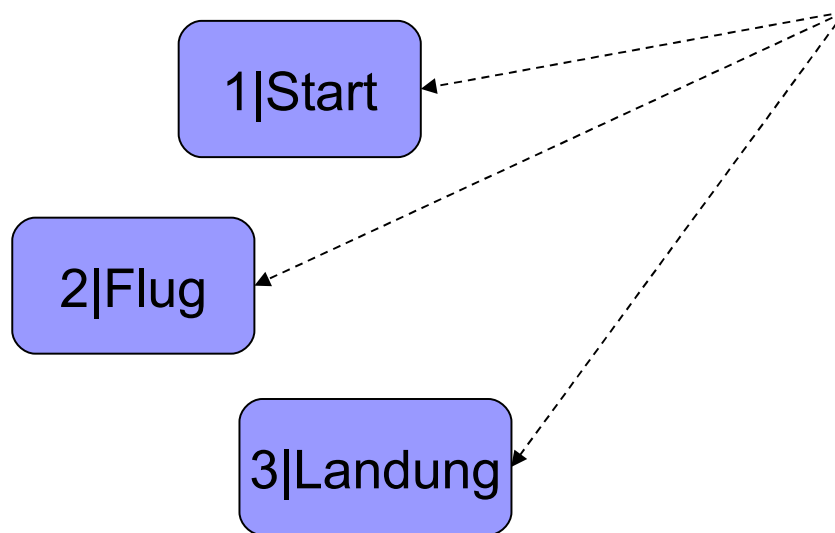
Ein eingebettetes System kann mehrere Arbeitsphasen durchlaufen.

Beispiel: Kontrollsystem eines Flugzeugs



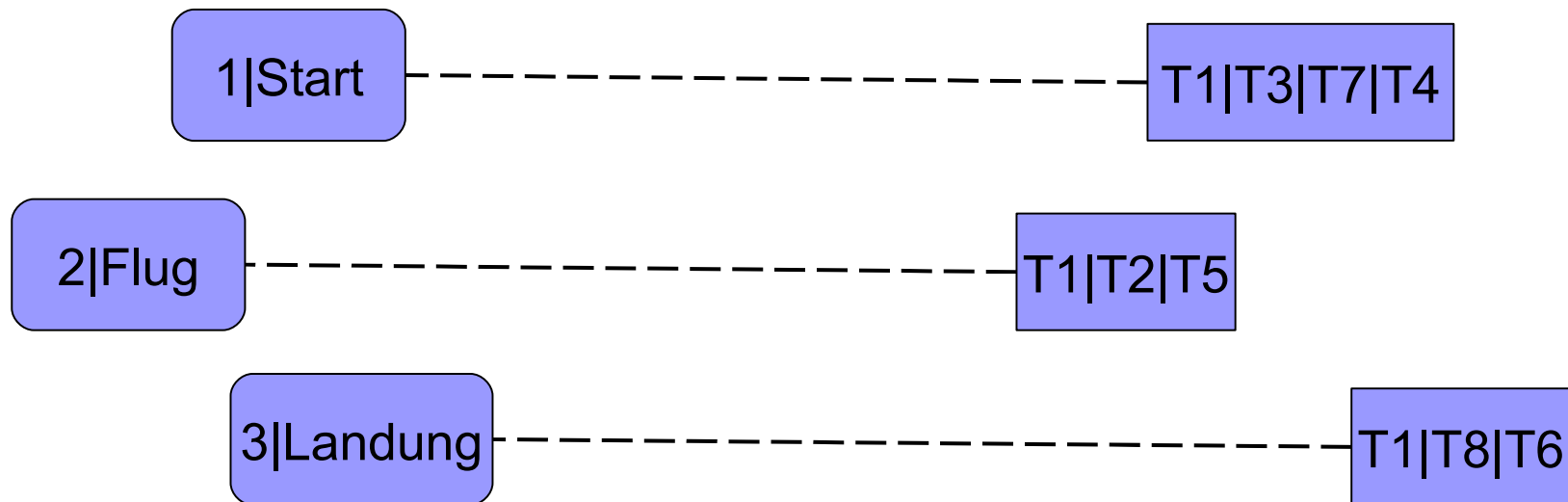
Wechsel des Arbeitsmodus

Phasen werden als Arbeitsmodi modelliert



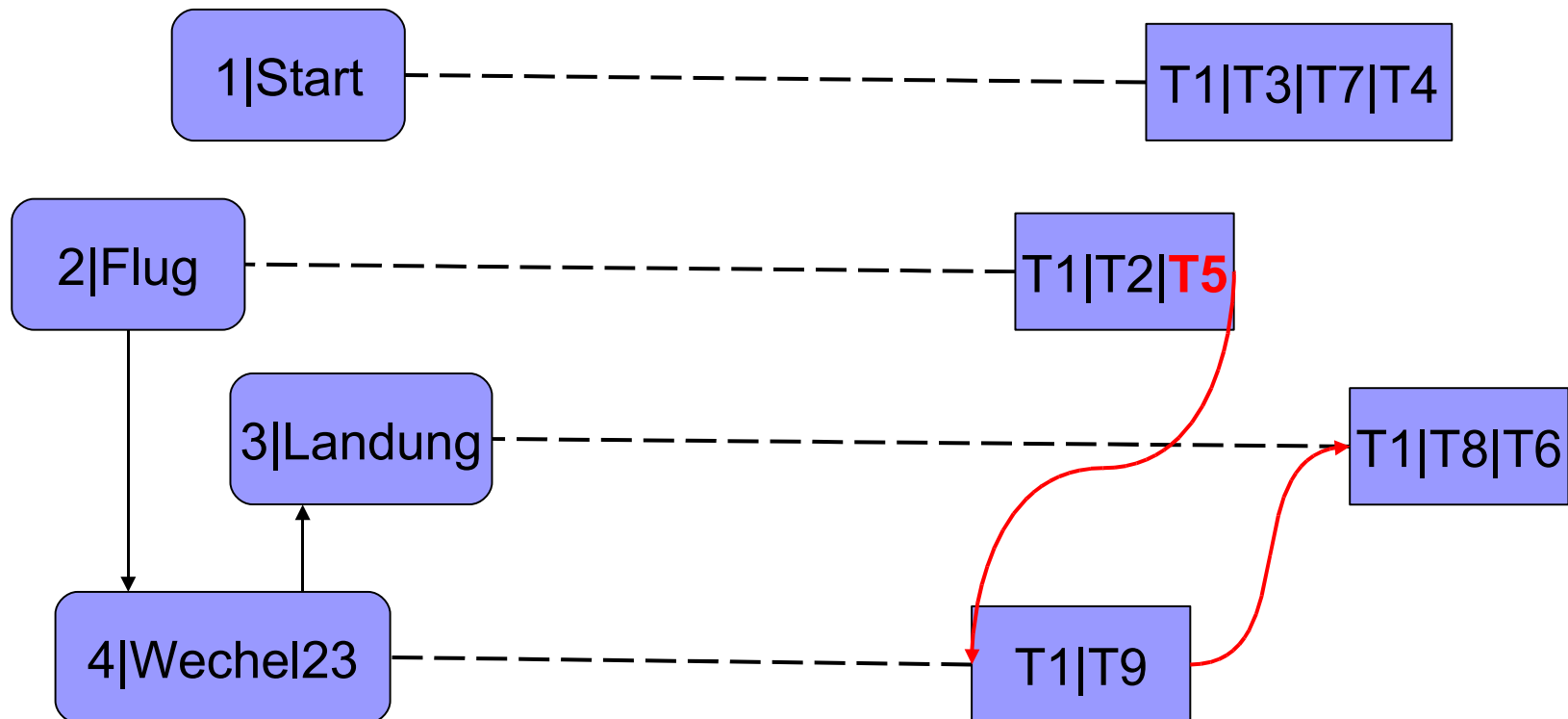
Wechsel des Arbeitsmodus

Jeder Arbeitsmodus besitzt einen eigenen statischen Ablaufplan.



Wechsel des Arbeitsmodus

Moduswechsel ausgelöst durch einen Task des aktuellen Modus



Wechsel des Arbeitsmodus

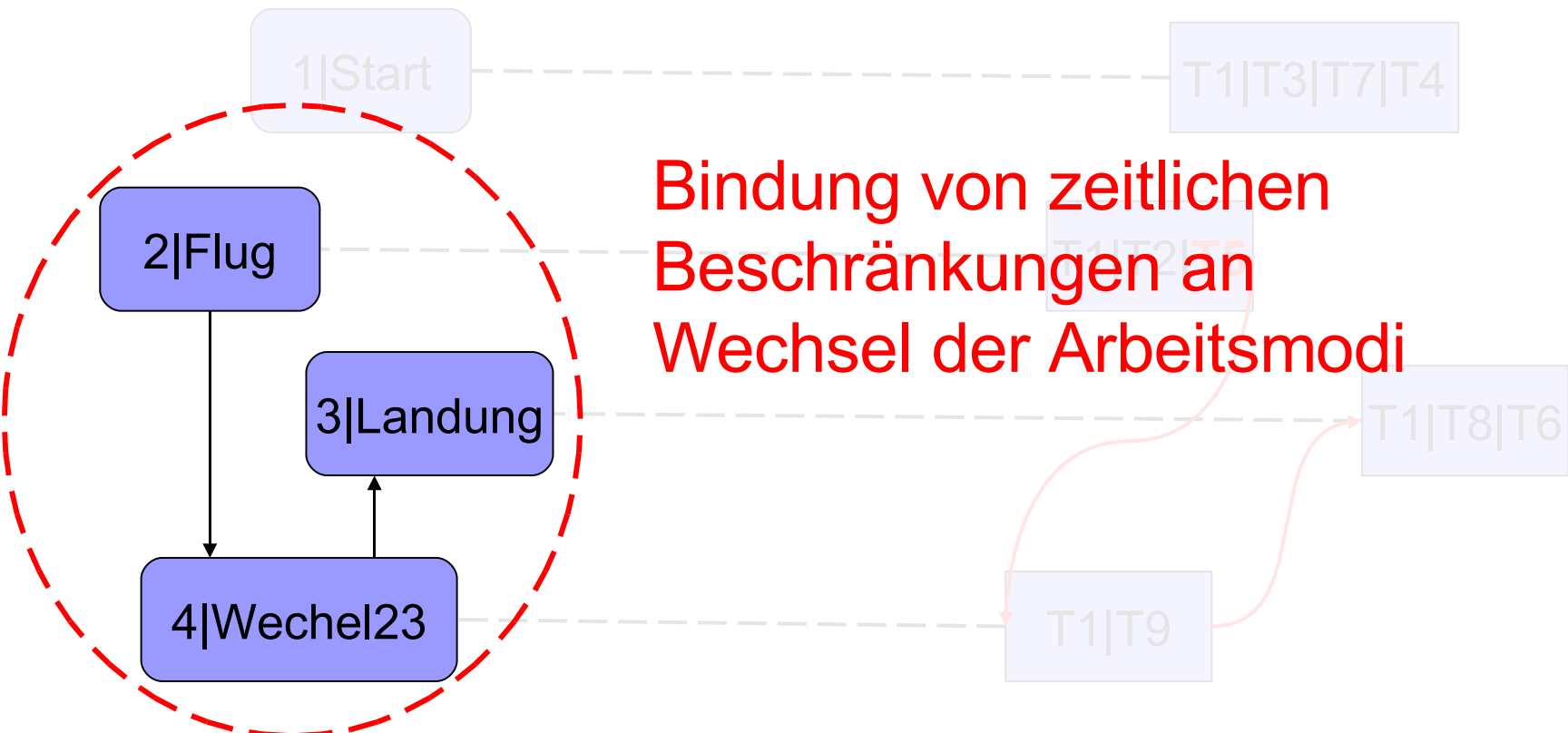
Arbeitsmodi können verschiedene zeitliche Anforderungen besitzen:

- Unkritischer Abschnitt
- Weiche Echtzeit
- Harte Echtzeit

Wechsel des Arbeitsmodus

Anforderungen für Echtzeitsysteme:

Zeitliches Verhalten muss deterministisch bleiben



Behandlung aperiodischer Tasks

Statische Planung folgender Aufgaben ist möglich:

- Natürlich-periodische Aufgaben
Aktivitäten von Natur aus periodisch
- Pseudo-periodische Aufgaben
Nicht periodisch, aber als solche modelliert

Behandlung aperiodischer Tasks

Erweiterung

Behandlung aperiodischer Ereignisse:

- Dynamisch während der Laufzeit
- Vorarbeit während der statischen Planung
→ *slot shifting*

Slot: Zeiteinheit, in der Tasks geplant werden

Behandlung aperiodischer Tasks

Grundidee:

Ungenutzte Zeitspannen in statischen Ablaufplänen werden für dynamische Aktivitäten benutzt.

Ablaufplan:



→ Kombination von statischer und dynamischer Ablaufplanung

Behandlung aperiodischer Tasks

Beispiel:

Task	Bereitzeit	Frist	Dauer
A	0	2	1
B	2	4	1

Ablaufplan:



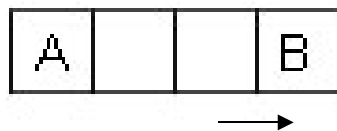
2 Slots der Größe 1 verfügbar:
zum Zeitpunkt 1 und 3

Behandlung aperiodischer Tasks

Beispiel:

Task	Bereitzeit	Frist	Dauer
A	0	2	1
B	2	4	1

Ablaufplanung mit *slot shifting*:



Ohne Verletzung der Frist kann Task B einen Task später geplant werden

Ergebnis:

1 Slot der Größe 2 ist zum Zeitpunkt 1 verfügbar!



**Vielen Dank für die
Aufmerksamkeit!**