



Technische
Universität
Braunschweig



DISPLIB Competition 2025 – The TRAInees

Abschlussprojekt AlgLab

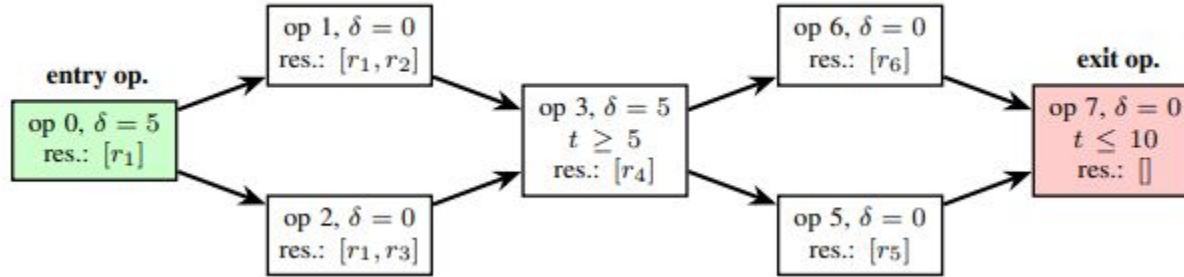
Gliederung

1. Einleitung
 - Problemstellung und Motivation
2. Eigenschaften der Instanzen
 - Eigenschaften und Besonderheiten
3. Mathematische Modellierung
 - Modellierung des Problems
4. Lösungsverfahren
 - Heuristik
 - Large Neighborhood Search (LNS) + Erweiterung (ALNS)
5. Ergebnisse
 - Lösungen der Instanzen
 - Platzierung bei der Competition
6. Ausblick auf Phase 2

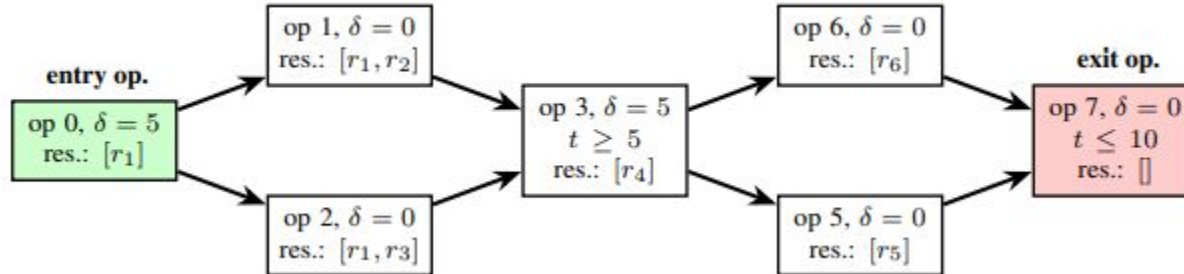
Problemstellung

- Bringe jeden Zug ins Ziel:

Zug1



Zug2

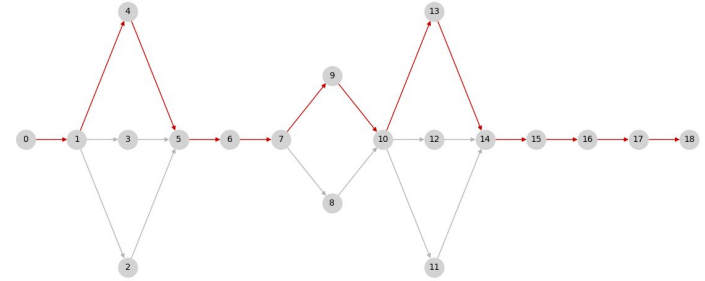


Eigenschaften und Besonderheiten der Instanzen

- Für Modellierung des Objective:
 - nur Increment oder Coeff treten für den selben Zug auf
 - Increment tritt nur mit Threshold 0 auf
- Für Heuristik
 - Züge in Full Instanzen blockieren für Start und Endoperationen keine Ressourcen

Modellierung des Problems – Path Constraints

- Operations-Graph für jeden Zug
 - Knoten \Leftrightarrow Operations
 - Kanten \Leftrightarrow Nachfolger-Relation
- Für jede Kante eine boolesche Variable
- Für jeden Zug gelten folgende Bedingungen:
 - Erste Operation: Summe der out-edges = 1
 - Letzte Operation: Summe der in-edges = 1
 - Alle anderen: Summe über in-edges = Summe über out-edges



Modellierung des Problems – Timing Constraints

- Jede Operation hat eine Integer Variable für Start und Ende
- Ist eine Operation ausgewählt, gilt:
 - $\text{Start}_{\text{op}} + \text{min_duration}_{\text{op}} \leq \text{Ende}_{\text{op}}$
 - $\text{Ende}_{\text{op}} = \text{Start}_{\text{succ}(\text{op})}$
 - $\text{Start}_{\text{op}} \leq \text{start_ub}_{\text{op}}$
 - $\text{start_lb}_{\text{op}} \leq \text{Start}_{\text{op}}$



Modellierung des Problems – Resource Constraints

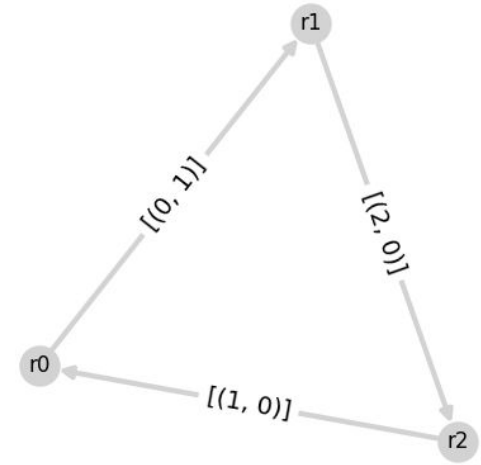
- Ermittle für jede Ressource, von welchen Operations sie genutzt wird
 - Beispiel: $r1 : [(train_1, op_2), (train_1, op_4), (train_3, op_1), \dots]$
- Für jede Ressource:
 1. Bilde für jede Operation eine optionale Intervall-Variable
 - $Start_{IV} \Leftrightarrow Start_{op}$
 - $Ende_{IV} \Leftrightarrow Ende_{op} + release_time_{resource}$
 - $optional \Leftrightarrow$ Wenn Operation ausgewählt ist
 2. Bilde jedes $(t_1, op_1), (t_2, op_2)$, das die Ressource verwendet:
 - Wenn $t \neq t' \Rightarrow$ fordere ein NoOverLap für die Intervallvariablen

Modellierung des Problems – Deadlock Constraints

- Resource-Allocation-Graph
 - Knoten \Leftrightarrow Ressourcen
 - Kanten \Leftrightarrow Ressourcen-Übergänge
- Verfahren:

Algorithm 1 Resource-Allocation-Graph

```
1:  $G \leftarrow \text{DiGraph}$ 
2: for each train do
3:   for each op do
4:     for each resource do
5:       if  $\text{release\_time}$  is 0 then
6:          $G.\text{add\_edge}(\text{res}_{op}, \text{res}_{succ(op)})$ 
7:       end if
8:     end for
9:   end for
10: end for
11: return  $G$ 
```



Resource-Allocation-Graph für die Test-Instanz swapping_2

Modellierung des Problems – Deadlock Constraints

- Wann genau wird eine Kante gezogen?
 - `release_time` der Ressource muss 0 sein \Rightarrow sonst keine Verklemmung möglich
 - Ressource der Folge-Operation muss eine andere sein
- Ein Kreis im Graph ist ein potenzieller Deadlock
 - Ein Kreis der Länge n benötigt mindestens n verschiedene Züge
- Nun:
 - Ersetze in der Instanz eine `release_time` von 0 mit boolescher Variable
 - Finde einen Zyklus der Länge n mit mind. n verschiedenen Zügen
 - $\text{Sum}(\text{release_time})$ entlang aller Kanten ≥ 1

→ Verklemmung ist gelöst

Modellierung des Problems – Threshold Constraints und Zielfunktion

- Hintergrund: Auswertung der Zielfunktion
 - Für je eine objective-component eine threshold-Variable
- Fall 1: $\text{Coeff}_{\text{component}} > 0$
 - Integer-Variable
 - $\text{var}_{\text{threshold}} \geq \text{start}_{\text{op}} - \text{component}_{\text{threshold}}$
- Fall 2: $\text{Increment}_{\text{component}} > 0$
 - boolesche Variable
 - $\text{start}_{\text{op}} + 1 \leq \text{component}_{\text{threshold}}$, wenn $\text{var}_{\text{threshold}} = \text{False}$
- Component einer Zielfunktion:
 - $\text{Increment}_{\text{component}} * \text{var}_{\text{threshold}} + \text{Coeff}_{\text{component}} * \text{var}_{\text{threshold}}$

Problem modelliert – Und nun?

- Ernüchternde Erkenntnis:
 - Schlechte Laufzeit
 - Keine (überzeugenden) Ergebnisse innerhalb von zehn Minuten
- Ursache:
 - Modell zu groß \Rightarrow zu viele Variablen und Constraints
- Neue Strategie:
 - Suche nicht nach globalen Optima
 - In kurzer Zeit gute Lösungen finden

Lösungsverfahren – Die Heuristik: Solver-Methode

- Gegeben:
 - Beliebige Instanz
- Schritt 1)
 - Jede `release_time`, die zuvor 0 war wird auf 1 gesetzt
- Schritt 2)
 - Wähle für jeden Zug genau eine zufällige Route
- Schritt 3)
 - Gib die erste zulässige Lösung zurück
- Enorme Zeitersparnis:
 - Path- und Deadlock-Constraints entfallen komplett
 - Anzahl an Variablen und Constraints wird deutlich reduziert

Lösungsverfahren – Die Heuristik: Algorithmische Methode

- Gegeben:
 - Instanz mit Zügen, die in ihrer ersten Operation keine Ressource belegen
 - Alle Operations abseits der Start-Operations haben keine *start_ub*
- Schritt 1)
 - Alle Züge starten ihre erste Operation zum Zeitpunkt 0
- Schritt 2)
 - Wähle für einen Zug eine zufällige Route
- Schritt 3)
 - Der nächste Zug startet, wenn:
 1. Sein Vorgänger seine Route beendet hat
 2. Zusätzlich die größte *release_time* vergangen ist

Lösungsverfahren – Die Heuristik : Kombination beider Strategien

- Bei vielen Instanzen ist es notwendig, beide Strategien zu kombinieren
- Schritt 1) Teile Instanz in zwei Teil-Instanzen A und B
 - A: Züge, die keine Ressourcen in ihrer ersten Operation verwenden
 - B: Züge, die mindestens eine Ressource in ihrer ersten Operation verwenden
- Schritt 2) Berechne jeweils eine zulässige Lösung für A und B
- Schritt 3) Füge die Teil-Lösungen zusammen
 - Teil-Instanz B fährt zuerst
 - Teil-Instanz A fährt danach
- Resultat:
 - Erhalte schnell zulässige Lösungen für jede Instanz



Lösungsverfahren – Large Neighborhood Search (LNS)

- Kernidee:
 - Eine zulässige Lösung iterativ verbessern
- Umsetzung:
 - Wähle eine Menge an Zügen als variabel
 - Die Restmenge an Zügen bleibt fix
 - Umplanung erfolgt mit einem Solver
- Vorteile
 - Weniger Variablen und Constraints
 - Schnelle Berechnung einer (besseren) Lösung

Lösungsverfahren – Model der LNS

	Fixe Züge	Variable Züge
Path-Constraints	Entfallen	Ja
Timing-Constraints	Entfallen	Ja
Resource-Constraints	Nur fixe Intervall-Variablen	Variable Intervall-Variablen
Deadlock-Constraints	Nur fixe Routen	Anpassung an fixe Züge
Threshold-Constraints	Entfallen	Ja

Lösungsverfahren – Adaptive Large Neighborhood Search (ALNS)

- Nach welcher Strategie wählen wir Züge zur Umplanung aus?
 - Least-used
 - Random
 - Highest objective
 - Most resource-conflicts
 - Nearest threshold
- Zuganzahl und Strategien haben Gewichte
 - Auswahl zufällig, jedoch nach Gewichtung
- Passe Gewichtung nach jeder Iteration an
 - Besseres Objective: Reward
 - Keine Verbesserung: Punishment

Ergebnisse – Lösungen der Instanzen

Instanz	Heuristik	Competition
critical_0	150.855	4.133
critical_1	75.964	2.416
critical_2	79.080	3.775
critical_3	363.135	8.584
critical_4	13.438	1.506
critical_5	39.377	2.677
critical_6	134.918	4.534
critical_7	103.991	4.316
critical_8	107.245	3.840
critical_9	118.462	5.514

Instanz	Heuristik	Competition
close_0	2.174	679
close_1	7.669	4.544
close_2	6.485	1.617
close_3	5.210	3.098
close_4	24.793	24.372
close_5	12.770	471
close_6	101.477	21.292
close_7	4.929	585
close_8	174.06	7.619

Instanz	Heuristik	Competition
full_0	22.881.938	7.024
full_1	33.901.797	12.922
full_2	2.680.044	10.640
full_3	5.430.667	3.518
full_4	13.621.449	11.551
full_5	26.080.804	8.773
full_6	33.217.196	9.741
full_7	32.332.985	6.863
full_8	37.318.418	11.467
full_9	35.344.136	12.652

Instanz	Heuristik	Competition
line_3_1	9.670	0
line_3_2	777.019.338	5.242
line_3_3	30.297.977	27.904
line_3_4	24.281.631	49.828
line_3_5	24.493.611	51.230

Instanz	Heuristik	Competition
headway_0	6.067	1.483
headway_1	8.395	5.040
headway_2	16.686	3.108
headway_3	7.826	3.552
headway_4	25.323	24.797
headway_5	14.093	869
headway_6	109.252	22.236
headway_7	11.549	1.150
headway_8	17.274	5.159

Instanz	Heuristik	Competition
headway_9	793.516.576	22.548
headway_10	47.477	9.809
headway_11	11.449	6.210

team	line1_critical	line1_full	line2_close	line2_headway	line3	SUM
openbus Florian Fuchs, Thomas Dubach, Jan Lordieck, Bernardo Martin-Iradi	76	95	80	77	42	370
CSLS Carolin Scholl, Luka Stärk	76	76	62	61	42	317
The TRAINees Lina Breuer, Sebastian Brunke, Elias Kaiser, Felix Michel	43	31	36	64	14	188
WUB (Wien, Udine, Bologna) Vera Hemmelmayr, Hai Yen Luu, Roberto Maria Rosati, Sara Ceschia, Andrea Schaeft, Alex Fabián Barrales Araneda, Valentina Cacchiani	76	10	45	4	4	139
Delayed Train Péter Dobrovoczkó, Péter Györgyi, Markó Horváth, Tamás Kis	8	50	20	30	18	126
TCS Research Railways team Aakash Swami, Gajendra Malviya	20	23	10	14	26	93
RailwayRam Hariharan Subramanian, Sachin Jayaswal	5	0	9	15	1	30
The A-Team Andres Espinoza, Alaina Gordon, Aleksandr Kazachkov, Yuyang Sun	0	0	0	0	0	0

Table 1: Points awarded for an instance.

Position	Phase 1	Phase 2
1st	10	15
2nd	7	11
3rd	5	8
4th	3	6
5th	2	4
6th	1	3
7th		2
8th		1

Ausblick zu Phase 2

- größere Instanzen
 - mehr Züge
 - mehr Operationen pro Zug
 - mehr Nachfolger
 - viele Züge der Phase 2 Instanzen blockieren in der Start- und Endoperation keine Ressourcen
- Ansätze für Future Work:
 - stärkere Aufteilung der Instanzen auf relevante Bereiche
 - Preprocessing von potentiellen Konflikten für Heuristik

Vielen Dank :)