Systematischer Vergleich heuristischer, approximativer und parametrisierter Algorithmen zur Routenoptimierung in der Lagerkommissionierung

Constantin Dietrich

Universität Freiburg
Lehrstuhl für Algorithmen und Datenstrukturen

23.09.2026

Gliederung

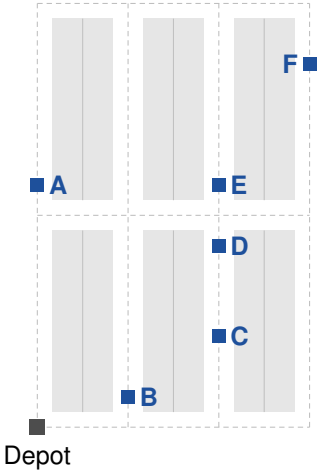
1 Problem
- 1.1 Kommissionierung
- 1.2 Modellierung und Definition
- 1.3 Komplexität und Forschungsfrage

2 Lösung
- 2.1 Nearest Neighbor
- 2.2 Christofides
- 2.3 Fixed-Parameter-Algorithmus
- 2.4 Webanwendung

3 Evaluation
- 3.1 Versuchsaufbau
- 3.2 Berechnungszeit
- 3.3 Lösungsqualität
- 3.4 Wirtschaftlichkeit

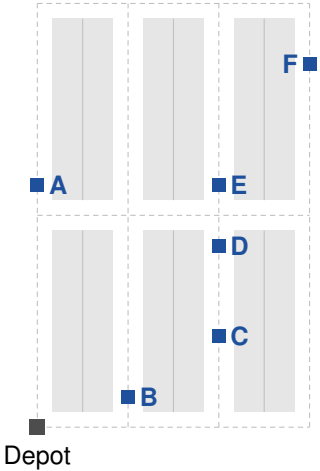
4 Fazit und Ausblick

1.1 Problem – Kommissionierung



- Ein Auftrag: n **Produkte** an verschiedenen Lagerorten (hier $n = 6$)
- Start und Ende am **Depot**

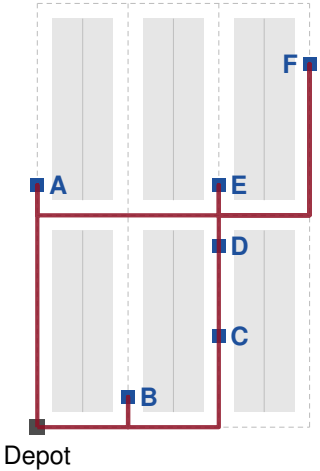
1.1 Problem – Kommissionierung



- Ein Auftrag: n **Produkte** an verschiedenen Lagerorten (hier $n = 6$)
- Start und Ende am **Depot**
- Gesucht: die kürzeste Route, die alle Produkte einsammelt (Order Picking Routing Problem, OPRP)

Kürzere Routen bedeuten kürzere Laufwege und damit geringere Kosten. Petersen 1997

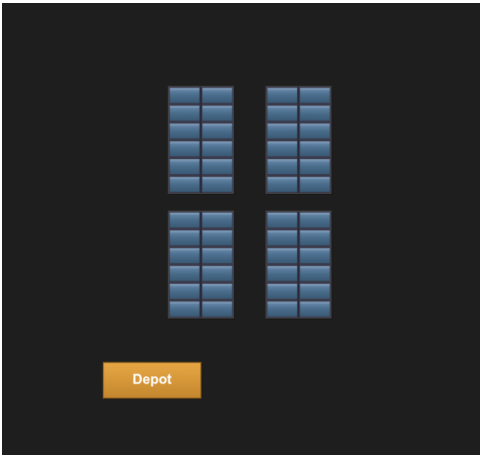
1.1 Problem – Kommissionierung



- Ein Auftrag: n **Produkte** an verschiedenen Lagerorten (hier $n = 6$)
- Start und Ende am **Depot**
- Gesucht: die kürzeste Route, die alle Produkte einsammelt (Order Picking Routing Problem, OPRP)
- Hier: optimale Route mit Länge 48

Kürzere Routen bedeuten kürzere Laufwege und damit geringere Kosten. Petersen 1997

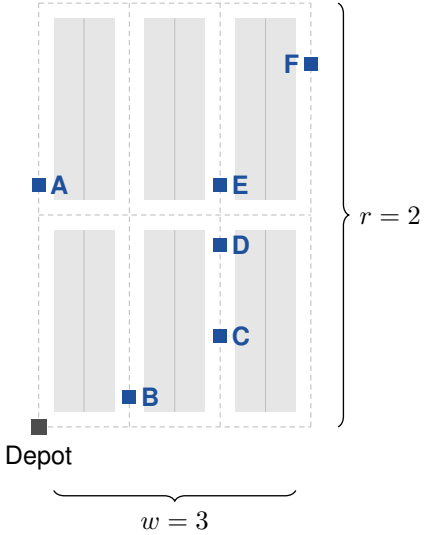
1.1 Problem – Demo



Webanwendung (Flask + JavaScript):

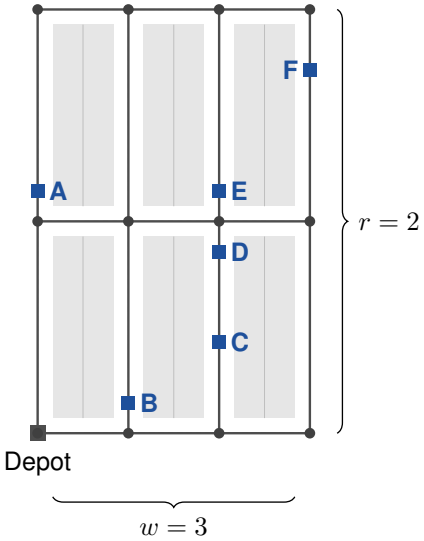
- Lagerlayout konfigurieren
- Auftrag zufällig erzeugen
- Routen der drei Verfahren berechnen und vergleichen

1.2 Problem – Das Lager als Graph



- Rechteckiges Lager mit w Regalspalten und r Regalreihen

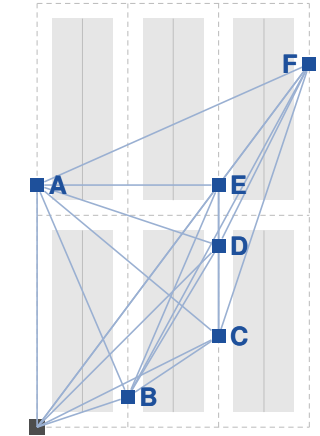
1.2 Problem – Das Lager als Graph



- Rechteckiges Lager mit w Regalspalten und r Regalreihen
- **Gittergraph**: Kreuzungen sind Knoten, Gänge sind Kanten
- Produkte sind *keine* Knoten, sondern Punkte *auf* einer Kante (Position im Gang)
- Kanten dürfen mehrfach begangen werden, Wenden im Gang ist erlaubt
- $h = r + 1$ Quergänge

Darauf rechnet das exakte Verfahren (Fixed-Parameter, Teil 2).

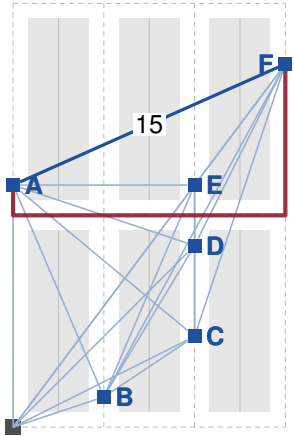
1.2 Problem – Der Distanzgraph



Nearest Neighbor (*Heuristik*) und Christofides (*Approximation*) brauchen nur die **Abstände** zwischen den Produkten.

- Knoten: Depot und die n Produkte
- Jedes Paar ist direkt verbunden

1.2 Problem – Der Distanzgraph



Nearest Neighbor (*Heuristik*) und Christofides (*Approximation*) brauchen nur die **Abstände** zwischen den Produkten.

- Knoten: Depot und die n Produkte
- Jedes Paar ist direkt verbunden
- Kantengewicht = Länge des kürzesten Laufwegs durch die Gänge, z. B. $d(A, F) = 15$

Die geraden Linien stehen nur für den Abstand, gelaufen wird immer durch die Gänge.

1.2 Problem – Definition

Order Picking Routing Problem (OPRP)

Gegeben: Gittergraph eines rechteckigen Lagers mit w Regalspalten und r Regalreihen, ein Depot und n Produktpositionen auf den Kanten

Gesucht: geschlossene Route ab dem Depot, die alle n Produkte besucht und minimale Länge hat

Ratliff & Rosenthal 1983, de Koster et al. 2007

Im Beispiel: $w = 3, r = 2, n = 6 \quad \Rightarrow \quad$ optimale Länge 48

1.3 Problem – Warum ist das schwer?

- Variante des **Travelling Salesperson Problem** (TSP)

1.3 Problem – Warum ist das schwer?

- Variante des **Travelling Salesperson Problem** (TSP)
- n Produkte ergeben $n!$ mögliche Reihenfolgen:

$$6! = 720 \qquad 30! \approx 2{,}7 \cdot 10^{32}$$

1.3 Problem – Warum ist das schwer?

- Variante des **Travelling Salesperson Problem** (TSP)
- n Produkte ergeben $n!$ mögliche Reihenfolgen:

$$6! = 720 \qquad 30! \approx 2{,}7 \cdot 10^{32}$$

- NP-schwer bei beliebig vielen Quergängen — Prunet et al. 2025
- Polynomiell für festes h: $O(n \cdot h \cdot 7^h)$ — Cambazard & Catusse 2018

1.3 Problem – Warum ist das schwer?

- Variante des **Travelling Salesperson Problem** (TSP)
- n Produkte ergeben $n!$ mögliche Reihenfolgen:

$$6! = 720 \qquad 30! \approx 2{,}7 \cdot 10^{32}$$

- NP-schwer bei beliebig vielen Quergängen — Prunet et al. 2025
- Polynomiell für festes h: $O(n \cdot h \cdot 7^h)$ — Cambazard & Catusse 2018
- In der Praxis: schnelle Heuristiken statt optimaler Routen

Trade-off: Berechnungszeit vs. Routenlänge

1.3 Problem – Forschungsfrage

Wie unterscheiden sich **Nearest Neighbor**, **Christofides** und der **Fixed-Parameter-Algorithmus** nach Cambazard & Catusse in Lösungsqualität und Laufzeit bei steigender Problemgröße und variierenden Lagerlayouts?

Fragen?

Fragen?

Gliederung

1 Problem

2 Lösung
- 2.1 Nearest Neighbor
- 2.2 Christofides
- 2.3 Fixed-Parameter-Algorithmus
- 2.4 Webanwendung

3 Evaluation

4 Fazit und Ausblick

2 Lösung – Drei Paradigmen

	Nearest Neighbor	**Christofides**	**Fixed-Parameter**
Paradigma	Heuristik	Approximation	parametrisiert, exakt
Garantie	keine konstante	$\leq 1{,}5 \cdot$ OPT	optimal
Laufzeit	$O(n^2)$	$O(n^3)$	$O(h \cdot v \cdot 7^h)$
rechnet auf	Distanzgraph	Distanzgraph	Gittergraph
Quelle	Rosenkrantz et al. 1977	Christofides 1976	Cambazard & Catusse 2018

n = Anzahl Produkte, h = Anzahl Quergänge, v = vertikale Gänge $(w+1)$, OPT = Länge der optimalen Route

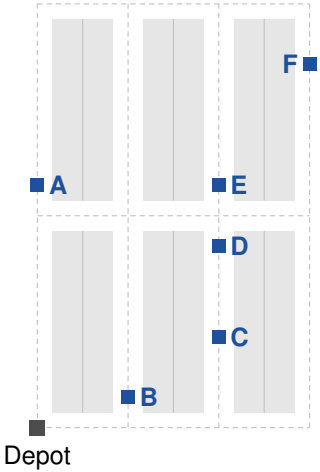
2 Lösung – Drei Paradigmen

	Nearest Neighbor	**Christofides**	**Fixed-Parameter**
Paradigma	Heuristik	Approximation	parametrisiert, exakt
Garantie	keine konstante	$\leq 1{,}5 \cdot$ OPT	optimal
Laufzeit	$O(n^2)$	$O(n^3)$	$O(h \cdot v \cdot 7^h)$
rechnet auf	Distanzgraph	Distanzgraph	Gittergraph
Quelle	Rosenkrantz et al. 1977	Christofides 1976	Cambazard & Catusse 2018

Jeweils ein Vertreter pro Paradigma.

n = Anzahl Produkte, h = Anzahl Quergänge, v = vertikale Gänge $(w+1)$, OPT = Länge der optimalen Route

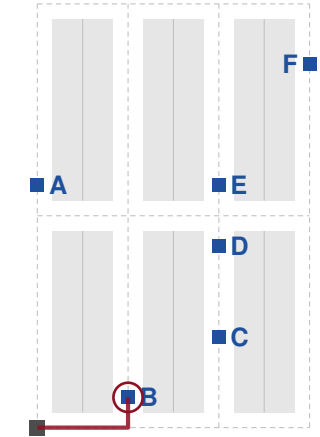
2.1 Lösung – Nearest Neighbor



- Starte am Depot
- **Greedy:** gehe immer zum **nächsten** noch offenen Produkt
- Zum Schluss zurück zum Depot

Rosenkrantz et al. 1977

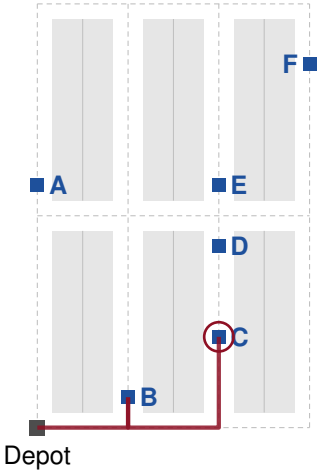
2.1 Lösung – Nearest Neighbor



- Starte am Depot
- **Greedy:** gehe immer zum **nächsten** noch offenen Produkt
- Zum Schluss zurück zum Depot

Rosenkrantz et al. 1977

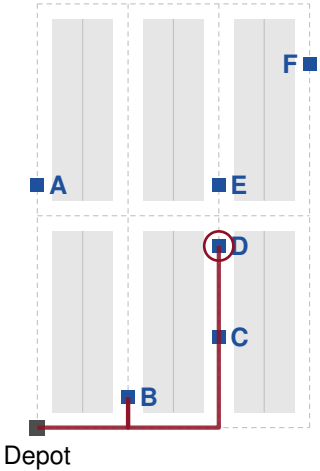
2.1 Lösung – Nearest Neighbor



- Starte am Depot
- **Greedy:** gehe immer zum **nächsten** noch offenen Produkt
- Zum Schluss zurück zum Depot

Rosenkrantz et al. 1977

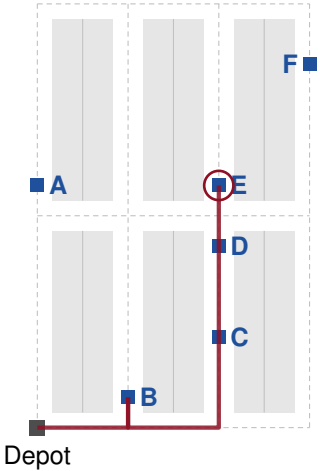
2.1 Lösung – Nearest Neighbor



- Starte am Depot
- **Greedy:** gehe immer zum **nächsten** noch offenen Produkt
- Zum Schluss zurück zum Depot

Rosenkrantz et al. 1977

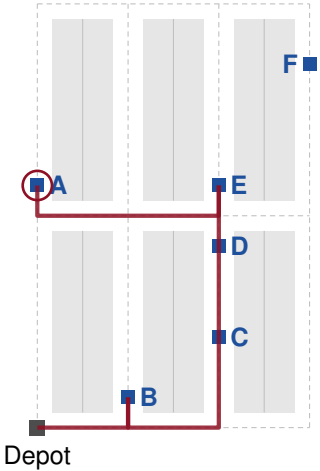
2.1 Lösung – Nearest Neighbor



- Starte am Depot
- **Greedy:** gehe immer zum **nächsten** noch offenen Produkt
- Zum Schluss zurück zum Depot

Rosenkrantz et al. 1977

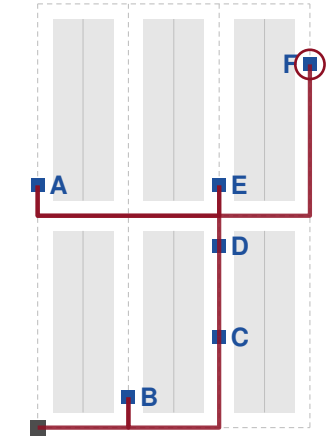
2.1 Lösung – Nearest Neighbor



- Starte am Depot
- **Greedy:** gehe immer zum **nächsten** noch offenen Produkt
- Zum Schluss zurück zum Depot

Rosenkrantz et al. 1977

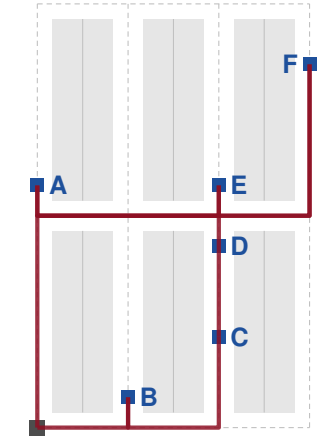
2.1 Lösung – Nearest Neighbor



- Starte am Depot
- **Greedy:** gehe immer zum **nächsten** noch offenen Produkt
- Zum Schluss zurück zum Depot

Rosenkrantz et al. 1977

2.1 Lösung – Nearest Neighbor



- Starte am Depot
- **Greedy:** gehe immer zum **nächsten** noch offenen Produkt
- Zum Schluss zurück zum Depot

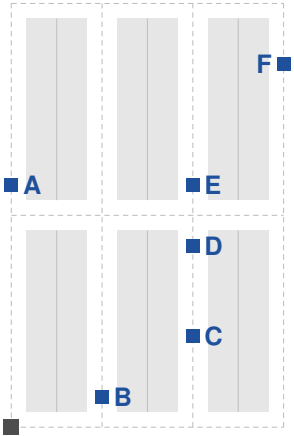
Reihenfolge: 0–B–C–D–E–A–F–0

Länge **60** statt 48 (+25 %)

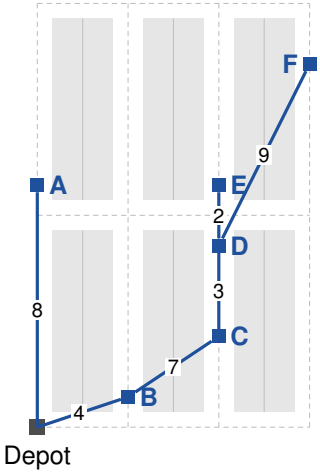
Der Umweg über A entsteht, weil früh nur lokal entschieden wird.

Rosenkrantz et al. 1977

2.2 Lösung – Christofides

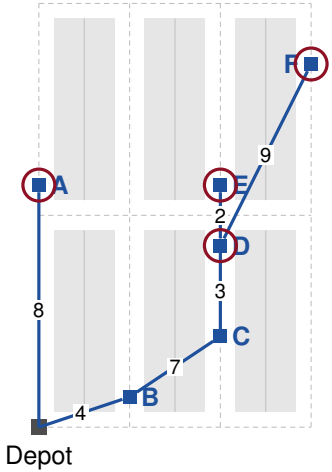


2.2 Lösung – Christofides



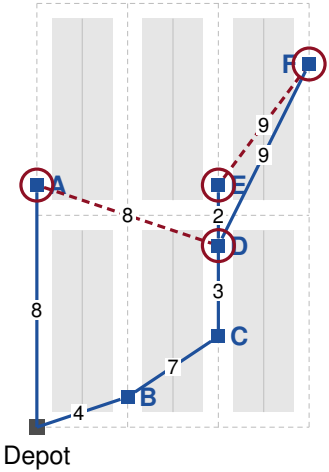
1. Minimaler Spannbaum (MST)

2.2 Lösung – Christofides



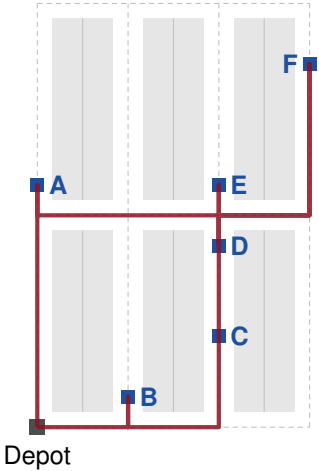
1. Minimaler Spannbaum (MST)
2. Knoten mit ungeradem Grad bestimmen

2.2 Lösung – Christofides



1. Minimaler Spannbaum (MST)
2. Knoten mit ungeradem Grad bestimmen
3. Perfektes Matching minimalen Gewichts darauf

2.2 Lösung – Christofides

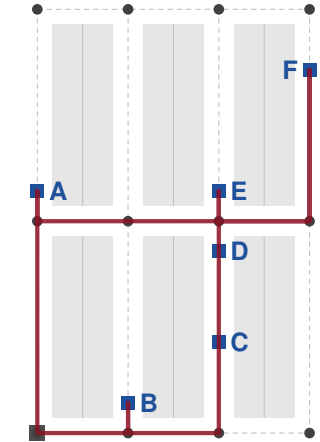


1. Minimaler Spannbaum (MST)
2. Knoten mit ungeradem Grad bestimmen
3. Perfektes Matching minimalen Gewichts darauf
4. Eulerkreis (jede Kante genau einmal), doppelte Knoten überspringen

Eulerkreis: 0–A–D–F–E–D–C–B–0
Route: 0–A–D–F–E–C–B–0

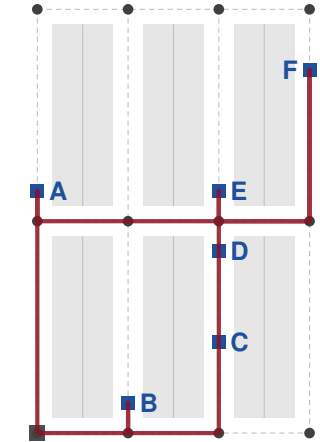
Länge **50** statt 48 (+4 %)

2.3 Lösung – Fixed-Parameter: Idee



Nicht die *Reihenfolge*, sondern:
Welche Gänge wie oft?

2.3 Lösung – Fixed-Parameter: Idee



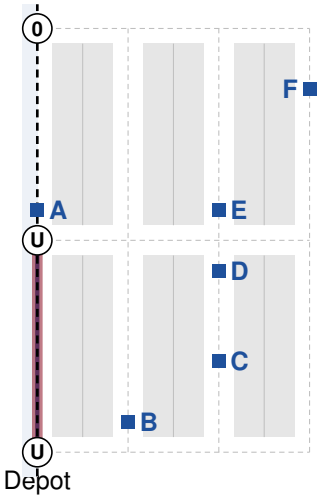
Nicht die *Reihenfolge*, sondern:
Welche Gänge wie oft?

Gültige Tour $\Leftrightarrow$

1. alle Produkte enthalten
2. zusammenhängend
3. alle Grade gerade

Ratliff & Rosenthal 1983

2.3 Lösung – Fixed-Parameter: Sweep

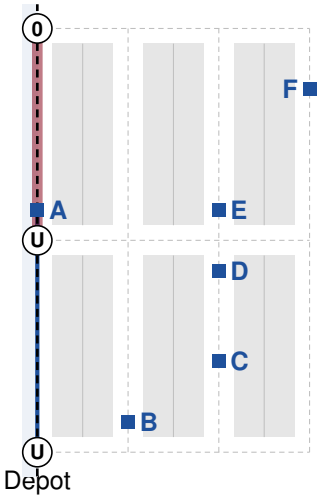


Spalte 1, Segment 1: **einmal durch**

Kreuzung	Parität	Komponente
oben	0	–
Mitte	U	1
unten	U	1

0 unbenutzt E gerade U ungerade

2.3 Lösung – Fixed-Parameter: Sweep

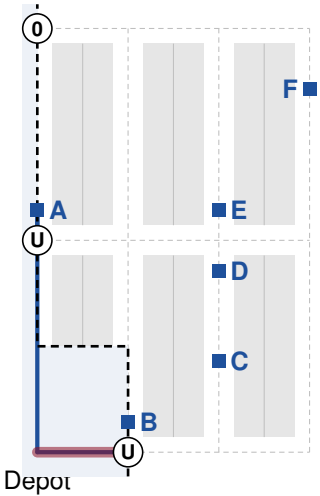


Spalte 1, Segment 2: **Stichgang von unten**

Kreuzung	Parität	Komponente
oben	0	–
Mitte	U	1
unten	U	1

0 unbenutzt E gerade U ungerade

2.3 Lösung – Fixed-Parameter: Sweep

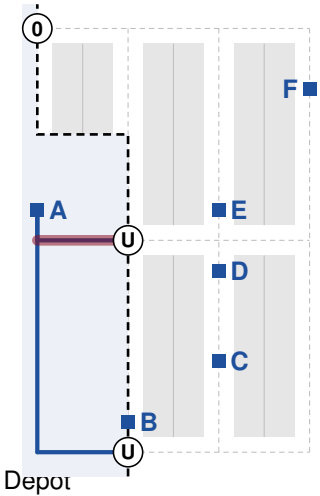


Quergang unten, Spalte 1 $\rightarrow$ 2: **einfach**

Kreuzung	Parität	Komponente
oben	0	–
Mitte	U	1
unten	U	1

0 unbenutzt E gerade U ungerade

2.3 Lösung – Fixed-Parameter: Sweep

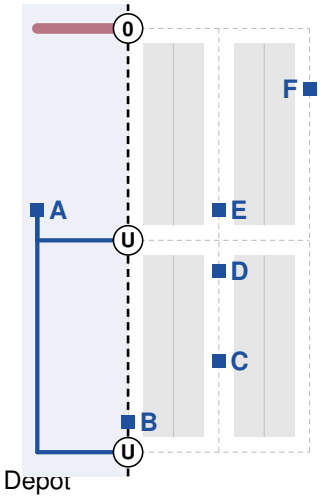


Quergang Mitte, Spalte 1 $\rightarrow$ 2: **einfach**

Kreuzung	Parität	Komponente
oben	0	–
Mitte	U	1
unten	U	1

0 unbenutzt E gerade U ungerade

2.3 Lösung – Fixed-Parameter: Sweep

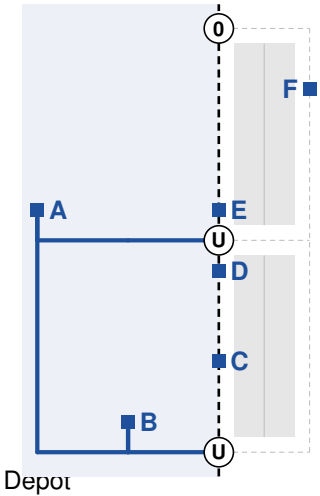


Quergang oben, Spalte 1 $\rightarrow$ 2: **nicht begangen**

Kreuzung	Parität	Komponente
oben	0	–
Mitte	U	1
unten	U	1

0 unbenutzt E gerade U ungerade

2.3 Lösung – Fixed-Parameter: Sweep

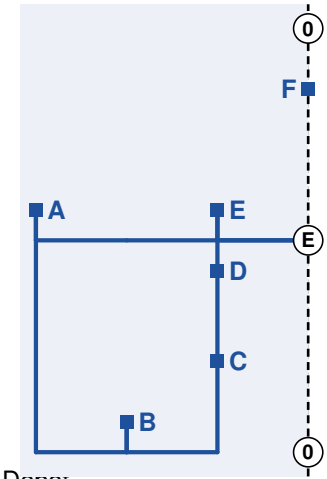


. . . Spalte 2 komplett, Frontier bei Spalte 3

Kreuzung	Parität	Komponente
oben	0	–
Mitte	U	1
unten	U	1

0 unbenutzt E gerade U ungerade

2.3 Lösung – Fixed-Parameter: Sweep

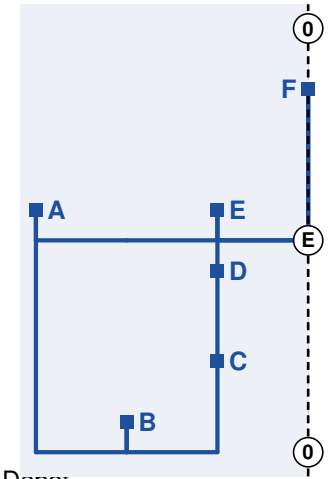


. . . Spalte 3 komplett, Frontier bei Spalte 4

Kreuzung	Parität	Komponente
oben	0	–
Mitte	E	1
unten	0	–

0 unbenutzt E gerade U ungerade

2.3 Lösung – Fixed-Parameter: Sweep

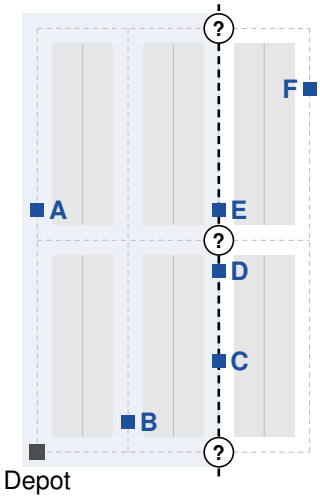


Letzte Spalte: **geschlossene Tour**, alle Paritäten gerade

Kreuzung	Parität	Komponente
oben	0	–
Mitte	E	1
unten	0	–

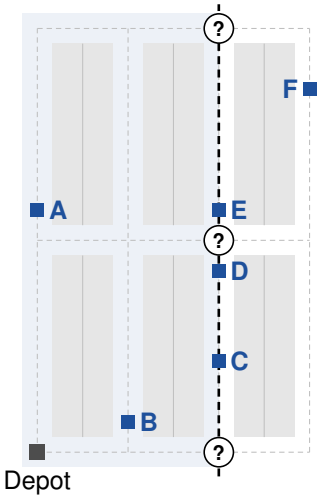
0 unbenutzt E gerade U ungerade

2.3 Lösung – Fixed-Parameter: Laufzeit



Zustand nur an den h **Kreuzungen**
(hier $h = 3$, unabhängig von n und Breite)

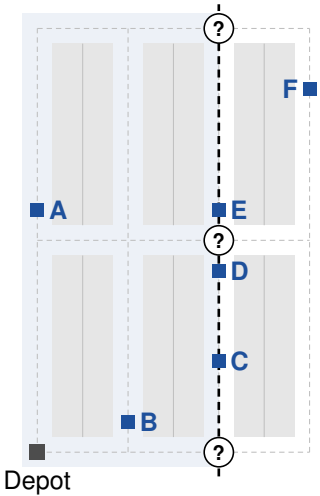
2.3 Lösung – Fixed-Parameter: Laufzeit



Zustand nur an den h **Kreuzungen**
(hier $h = 3$, unabhängig von n und Breite)

$\Rightarrow$ höchstens $O(7^h)$ Zustände

2.3 Lösung – Fixed-Parameter: Laufzeit



Zustand nur an den h **Kreuzungen**
(hier $h = 3$, unabhängig von n und Breite)

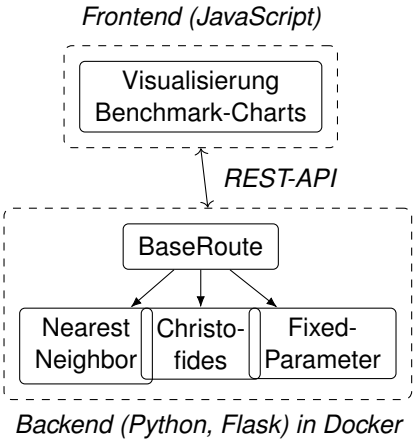
$\Rightarrow$ höchstens $O(7^h)$ Zustände

$$O(h \cdot v \cdot 7^h)$$

linear in der Lagergröße, exponentiell in $h = r + 1$

$v = w + 1$ vertikale Gänge

2.4 Lösung – Webanwendung und Benchmarks



- Alle drei Verfahren auf demselben Lagermodell
- Jeder Lauf in eigenem Prozess
- Gemessen wird nur die Routenberechnung
- Timeout 90 s
- Reproduzierbar über Seeds und Docker

Fragen?

Fragen?

Gliederung

1 Problem

2 Lösung

3 Evaluation

3.1 Versuchsaufbau
3.2 Berechnungszeit
3.3 Lösungsqualität
3.4 Wirtschaftlichkeit

4 Fazit und Ausblick

3.1 Evaluation – Versuchsaufbau

Drei Benchmarks

- B1: Breite $w = 1 \dots 10$, $r = 1$
- B2: Regalreihen $r = 1 \dots 10$, $w = 10$
- B3: quadratisch $1 \times 1 \dots 10 \times 10$

Je Layout

- $n = 5, 10, \dots, 30$ Produkte
- 20 zufällige Aufträge

Fest: Regalblock 2×6 Lagerplätze, Depot unten links

Insgesamt 10.200 Läufe

Gemessen

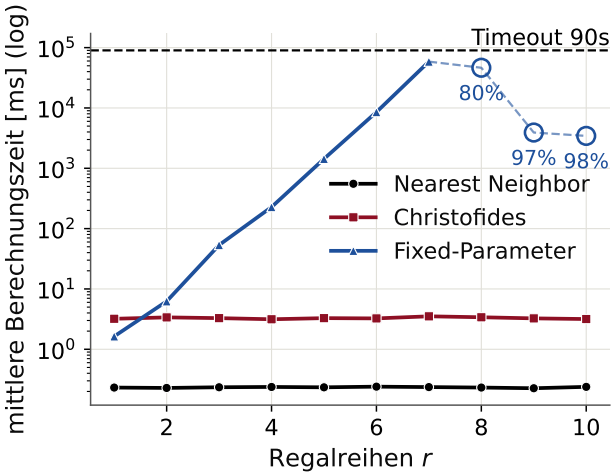
- Berechnungszeit
- Routenlänge

Optimalitätsgap

$$\text{Gap} = \frac{L_{\text{Algorithmus}} - L_{\text{Optimum}}}{L_{\text{Optimum}}}$$

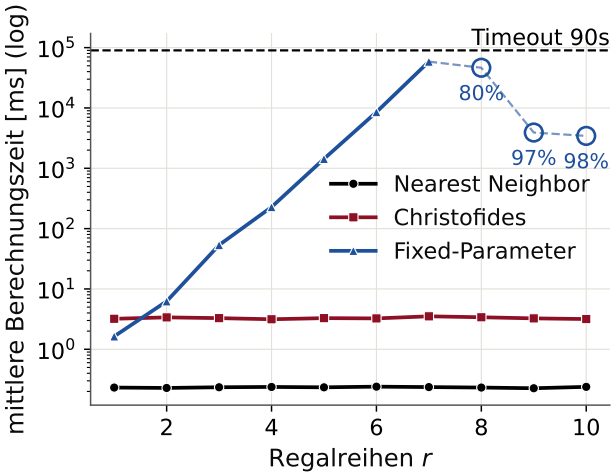
L_{Optimum} vom Fixed-Parameter-Algorithmus

3.2 Evaluation – Berechnungszeit (B2)



- Nearest Neighbor: $\approx 0{,}2$ ms
- Christofides: ≈ 3 ms
- beide layoutunabhängig – ihre Zeit bestimmt n ($\times 26$ von 5 auf 30 Produkte)
- Fixed-Parameter: wächst pro Regalreihe um Faktor 4–9

3.2 Evaluation – Berechnungszeit (B2)



- Nearest Neighbor: $\approx 0{,}2$ ms
- Christofides: ≈ 3 ms
- beide layoutunabhängig – ihre Zeit bestimmt n ($\times 26$ von 5 auf 30 Produkte)
- Fixed-Parameter: wächst pro Regalreihe um Faktor 4–9
- $r = 7$: 58,6 s im Mittel
- $r \geq 8$: 80–98 % Timeouts

3.3 Evaluation – Lösungsqualität

Optimalitätsgap gegenüber der exakten Lösung (nur Layouts ohne Timeout)

	Nearest Neighbor		Christofides	
Benchmark	Mittel	Max	Mittel	Max
B1 (Breite)	4,1 %	35,7 %	3,6 %	15,8 %
B2 (Regalreihen)	11,6 %	44,4 %	7,5 %	23,5 %
B3 (quadratisch)	13,7 %	50,0 %	7,9 %	24,2 %

3.3 Evaluation – Lösungsqualität

Optimalitätsgap gegenüber der exakten Lösung (nur Layouts ohne Timeout)

	Nearest Neighbor		Christofides	
Benchmark	Mittel	Max	Mittel	Max
B1 (Breite)	4,1 %	35,7 %	3,6 %	15,8 %
B2 (Regalreihen)	11,6 %	44,4 %	7,5 %	23,5 %
B3 (quadratisch)	13,7 %	50,0 %	7,9 %	24,2 %

- Abstand wächst mit mehr Regalreihen und größeren Aufträgen:
  bei $r = 7$ steigt der NN-Gap von 5,3 % (5 Produkte) auf 23,3 % (30 Produkte)
- Christofides bleibt weit unter seiner Garantie von 50 %

3.4 Evaluation – Lohnt sich die exakte Lösung?

Eingesparte Gehzeit $>$ zusätzliche Rechenzeit? (1 m/s, 1 Gittereinheit = 1 m)

r	gesparte Gehzeit	Mehrrechenzeit	lohnt sich?
1	4,4 s	$< 0{,}01$ s	ja
6	16,0 s	8,5 s	ja
7	20,1 s	58,6 s	nein

Fixed-Parameter gegenüber Christofides, Mittelwerte aus B2

3.4 Evaluation – Lohnt sich die exakte Lösung?

Eingesparte Gehzeit $>$ zusätzliche Rechenzeit? (1 m/s, 1 Gittereinheit = 1 m)

r	gesparte Gehzeit	Mehrrechenzeit	lohnt sich?
1	4,4 s	$< 0{,}01$ s	ja
6	16,0 s	8,5 s	ja
7	20,1 s	58,6 s	nein

Fixed-Parameter gegenüber Christofides, Mittelwerte aus B2

Wirtschaftliche Grenze zwischen $r = 6$ und $r = 7$

3.4 Evaluation – Lohnt sich die exakte Lösung?

Eingesparte Gehzeit $>$ zusätzliche Rechenzeit? (1 m/s, 1 Gittereinheit = 1 m)

r	gesparte Gehzeit	Mehrrechenzeit	lohnt sich?
1	4,4 s	$< 0{,}01$ s	ja
6	16,0 s	8,5 s	ja
7	20,1 s	58,6 s	nein

Fixed-Parameter gegenüber Christofides, Mittelwerte aus B2

Wirtschaftliche Grenze zwischen $r = 6$ und $r = 7$

Bei $r = 6$ entscheidet der Auftrag:

5 Produkte: 3,8 s gespart $<$ 4,3 s Rechenzeit $\Rightarrow$ nein

ab 10 Produkten: 9,7–22,9 s gespart $>$ 6,5–10,8 s $\Rightarrow$ ja

3.4 Evaluation – Entscheidungsmatrix

	Rechenzeit	Zeitbudget pro Auftrag	
r	Fixed-Parameter	≈ 5 ms	≥ 10 s
≤ 5	bis 1,4 s	Nearest Neighbor	Fixed-Parameter
6	8,5 s	Nearest Neighbor	Fixed-Parameter ab $n = 10$
≥ 7	ab 58,6 s	Nearest Neighbor	Christofides

Mittelwerte aus B2 ($w = 10$), $n = 5 \ldots 30$; bei $r = 6$ und $n < 10$ Christofides

4 Fazit

- Exakt: berechenbar bis $r = 7$, lohnt sich bis $r = 5$ ($r = 6$ ab 10 Produkten)
- Sonst Christofides: 3,6–7,9 % über dem Optimum
  Nearest Neighbor: 4,1–13,7 %
- Beiträge: Dreiervergleich, Benchmark-Werkzeug, Entscheidungsmatrix

4 Fazit

- Exakt: berechenbar bis $r = 7$, lohnt sich bis $r = 5$ ($r = 6$ ab 10 Produkten)
- Sonst Christofides: 3,6–7,9 % über dem Optimum
  Nearest Neighbor: 4,1–13,7 %
- Beiträge: Dreiervergleich, Benchmark-Werkzeug, Entscheidungsmatrix

Ausblick: kompilierte Implementierung mit Preprocessing, Vergleich mit S-Shape und Largest-Gap

Fragen?

Fragen?

Quellen

Cambazard, H. und Catusse, N. (2018). Fixed-Parameter Algorithms for Rectilinear Steiner Tree and Rectilinear Traveling Salesman Problem in the Plane. *European Journal of Operational Research* 270(2), 419–429.

Christofides, N. (1976). Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem. Report 388, Carnegie Mellon University.

Petersen, C. G. (1997). An Evaluation of Order Picking Routeing Policies. *International Journal of Operations & Production Management* 17(11), 1098–1111.

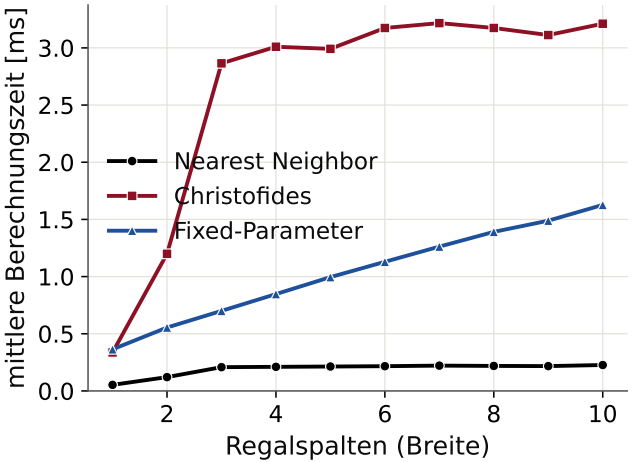
Prunet, T., Absi, N. und Cattaruzza, D. (2025). A Note on the Complexity of the Picker Routing Problem in Multi-Block Warehouses and Related Problems. *Annals of Operations Research* 347, 1595–1605.

Ratliff, H. D. und Rosenthal, A. S. (1983). Order-Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem. *Operations Research* 31(3), 507–521.

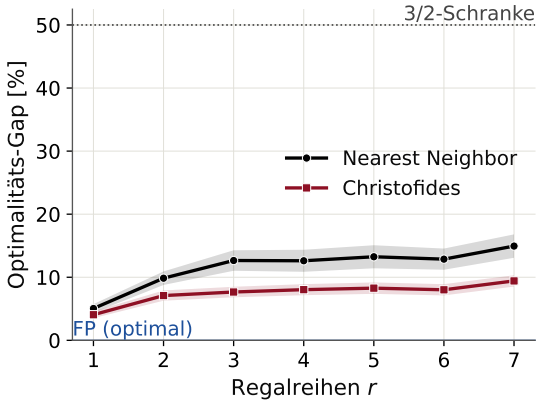
Rosenkrantz, D. J., Stearns, R. E. und Lewis, P. M. (1977). An Analysis of Several Heuristics for the Traveling Salesman Problem. *SIAM Journal on Computing* 6(3), 563–581.

Backup

Backup – Einfluss der Lagerbreite (B1)

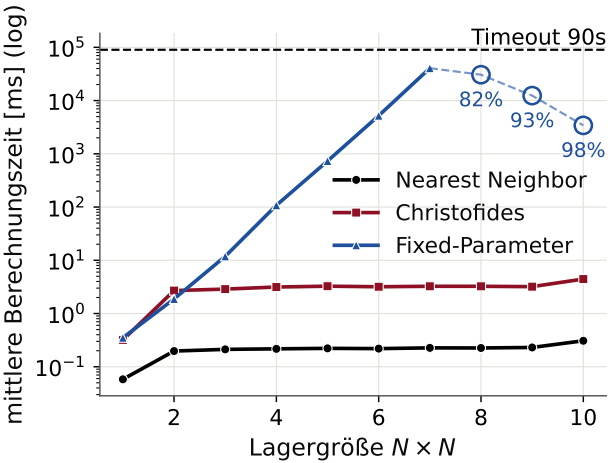


Backup – Gap über Regalreihen (B2)

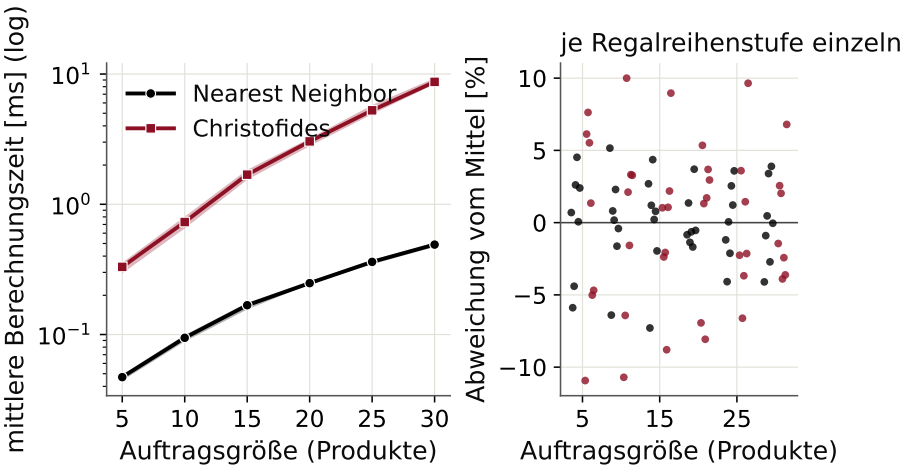


Keine verlässliche FP-Referenz (Timeouts) für: 8, 9, 10

Backup – Quadratische Layouts (B3)



Backup – Einfluss der Auftragsgröße



Backup – Statistische Absicherung

- 20 Wiederholungen je Konfiguration mit festen Seeds (42–61)
- Wilcoxon-Vorzeichen-Rang-Test über Einzelinstanzen
  - B2, B3: Christofides signifikant besser ($p < 10^{-30}$)
  - B1: Unterschied nicht signifikant ($p = 0{,}068$)
- 840 doppelt gemessene Läufe stimmen in Status und Routenlänge überein

Backup – Warum $\leq 1{,}5 \cdot \mathrm{OPT}$?

- OPT ohne eine Kante ist ein Spannbaum $\Rightarrow$ MST $\leq$ OPT
- OPT nur auf den ungeraden Knoten abgekürzt ist ein Kreis gerader Länge und zerfällt in 2 Matchings $\Rightarrow$ Matching $\leq \frac{1}{2}$ OPT
- Eulerkreis abkürzen wird nicht länger (Dreiecksungleichung) $\Rightarrow \leq \frac{3}{2}$ OPT

Backup – Expansionsmöglichkeiten

Gangsegment (bis zu 6 Möglichkeiten)

auslassen
0

einmal
L

doppelt
$2L$

Stich unten
$2d_u$

Stich oben
$2d_o$

beidseitig
$2L - 2\gamma$

Quergang (nach Parität)

$U \Rightarrow$ weiter
$0, E \Rightarrow$ abhängen oder doppelt

Backup – Expansionsmöglichkeiten

Gangsegment (bis zu 6 Möglichkeiten)

auslassen
0

einmal
L

doppelt
$2L$

Stich unten
$2d_u$

Stich oben
$2d_o$

beidseitig
$2L - 2\gamma$

Quergang (nach Parität)

$U \Rightarrow$ weiter
$0, E \Rightarrow$ abhängen oder doppelt

Pro Zustand nur die günstigste Teiltour

Backup – Limitationen

- Ein Layouttyp: rechteckig, parallele Regale, Depot in der Ecke
- Feste Regalmaße, Auftragsgröße nur 5–30 Produkte
- Gleichverteilte Lagerorte, eine Kommissionierkraft
- Keine exakte Referenz für $r \geq 8$ (Survivorship-Bias der Läufe ohne Timeout)
- Python-Implementierung ohne Preprocessing