



UNIVERSITÄT  
LEIPZIG

Masterarbeit

**ENTWICKLUNG EINES ACO BASIERTEN ALGORITHMUS  
ZUR LÖSUNG EINES MULTIKRITERIELLEN  
OPTIMIERUNGSPROBLEMS IN DER AUFTRAGSPLANUNG**

Leipzig, 10.06.2021

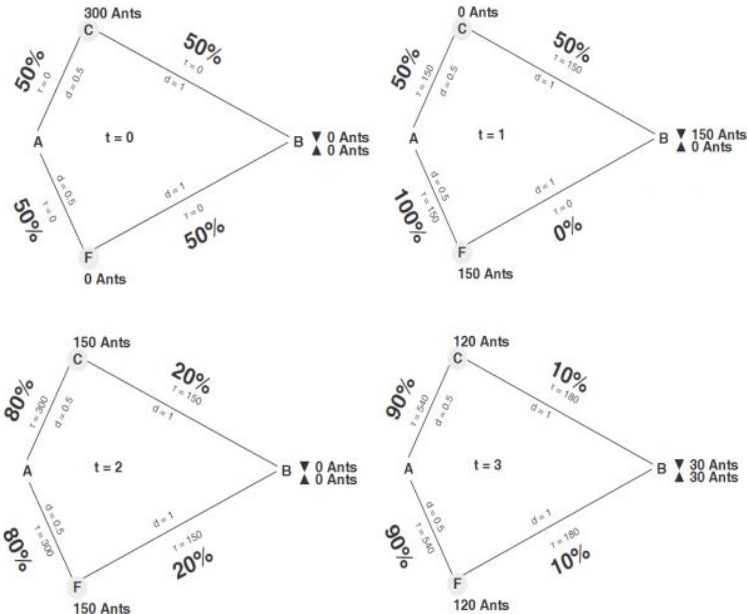
Falk Müller

# GLIEDERUNG

1. Grundlagen
2. Aufgabe
3. Umsetzung
4. Messung
5. Lokale Suche
6. Ausblick

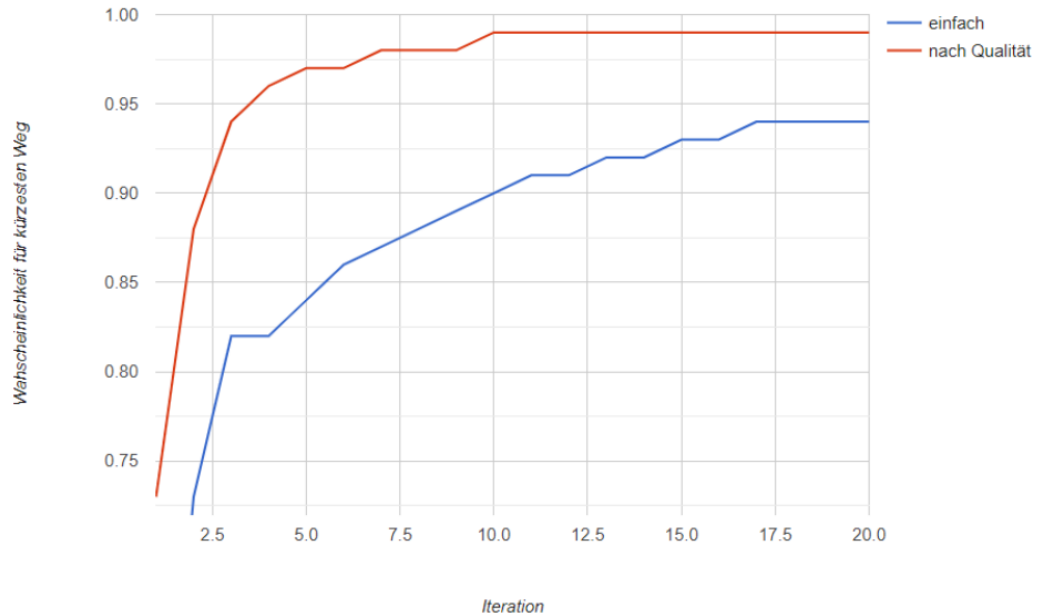
# GRUNDLAGEN

# AMEISENALGORITHMEN

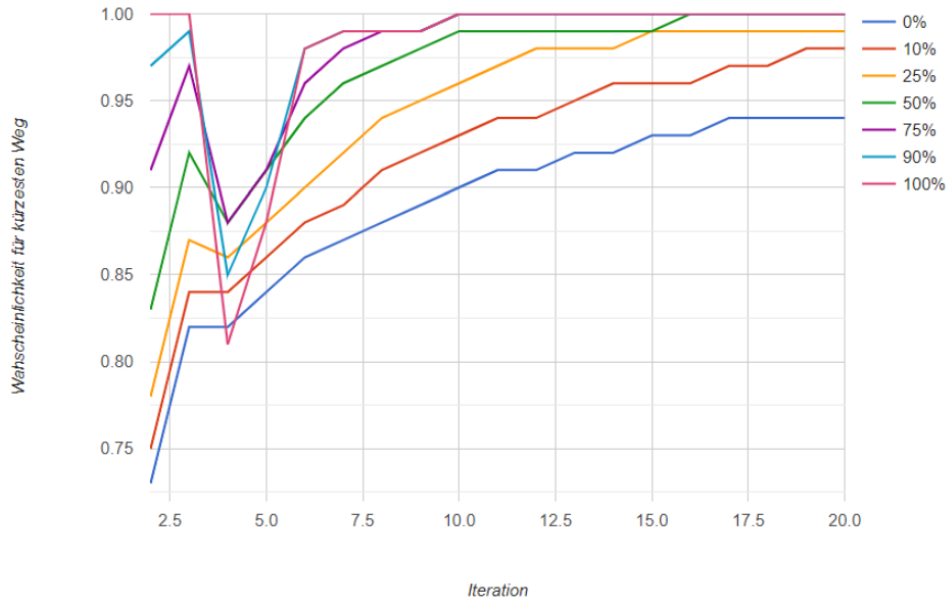


[https://code.falk-m.de/demo\\_ants.html](https://code.falk-m.de/demo_ants.html)

# PHEROMON-UPDATE



# VERDUNSTUNG



$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}$$

# HEURISTISCHE ANTEIL

$$p_{ij} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{j \in \text{mögliche Knoten}} \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}$$

# AUFGABE



# AUFTRAGSVERTEILUNG

- 200 Bearbeiter
- 3000 Aufträge

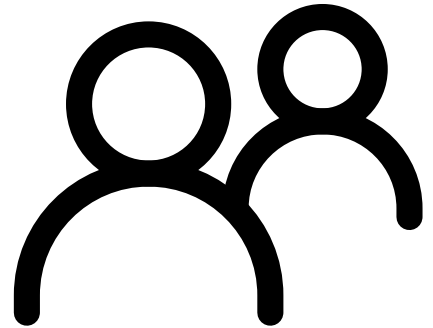


## AUFTRÄGE

- $P_j$ : geografische Position
- $DUE_j$ : Abgabedatum
- $PR_j$ : Bearbeitungsdauer vor Ort
- $TC_j$ : Verspätungskosten je Zeiteinheit
- $PROB_j$ : Liste der obligatorische Eigenschaften
- $PROP_j$ : Liste der optionalen Eigenschaften

## BEARBEITER

- $PM_i$ : geografische Position
- $CPH_i$ : Stundenlohn
- $PRO_i$ : Liste von Eigenschaften



## FUNKTIONEN

- **getDistance**(i,j)
- **getTravilingTime**(i,j)  $\rightarrow$  getDistance(i,j) \* HPK
- **GetTardinessCosts**(i, t)  $\rightarrow$  MAX( $TC_j * (t - DUE_j + PR_j)$ , 0)
- **getStaffOrderCosts**(i, j)  $\rightarrow$  CPH<sub>i</sub> \* PR<sub>j</sub>
- **getStaffTravelCosts**(t, i)  $\rightarrow$  CPH<sub>i</sub> \* t
- **getStaffOrderPreference**(i, j)  
 $\rightarrow (|PROB_j \cap PRO_i| + |PROP_j \cap PRO_i|) / (|PROB_j| + |PROP_j|)$ , wenn  $|PROB_j| + |PROP_j| > 0$   
 $\rightarrow 0.5$ , wenn  $|PROB_j| + |PROP_j| = 0$

# KOSTEN

- KFZ: Benzin-Kosten
- Verspätungskosten
- Reisekosten
- Bearbeitungskosten
- Unzufriedenheit



## ZIEL

- MIN(getTotalCosts(R))

getTotalCosts(R)=

$$\sum_{i=0}^{|M|} \text{getTourTardinessCosts}(R_i, i) + \sum_{j \in R_i} \left( \begin{aligned} &\text{getDistance}(i, j) * 0.1 \\ &+ \text{getStaffOrderCosts}(i, j) \\ &+ (1 - \text{getStaffOrderPreference}(i, j)) * 100 \\ &+ \text{getStaffTravelCosts}(i, \text{getTravilingTime}(i, j)) \end{aligned} \right)$$

# UMSETZUNG

## ALG 1

- Bestimmung Reihenfolge
  - Eigene Pheromonmatrix
- Zuordnung zum Bearbeiter
  - Eigene Pheromonmatrix

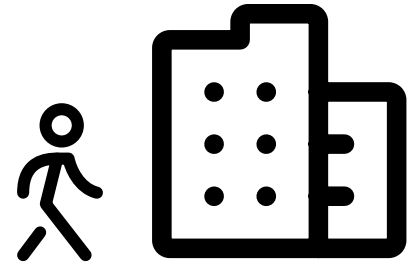
## ALG 2

- Bestimmung Zuordnung zum Bearbeiter
- Reihenfolge je Bearbeiter
  - Matrix je Bearbeiter

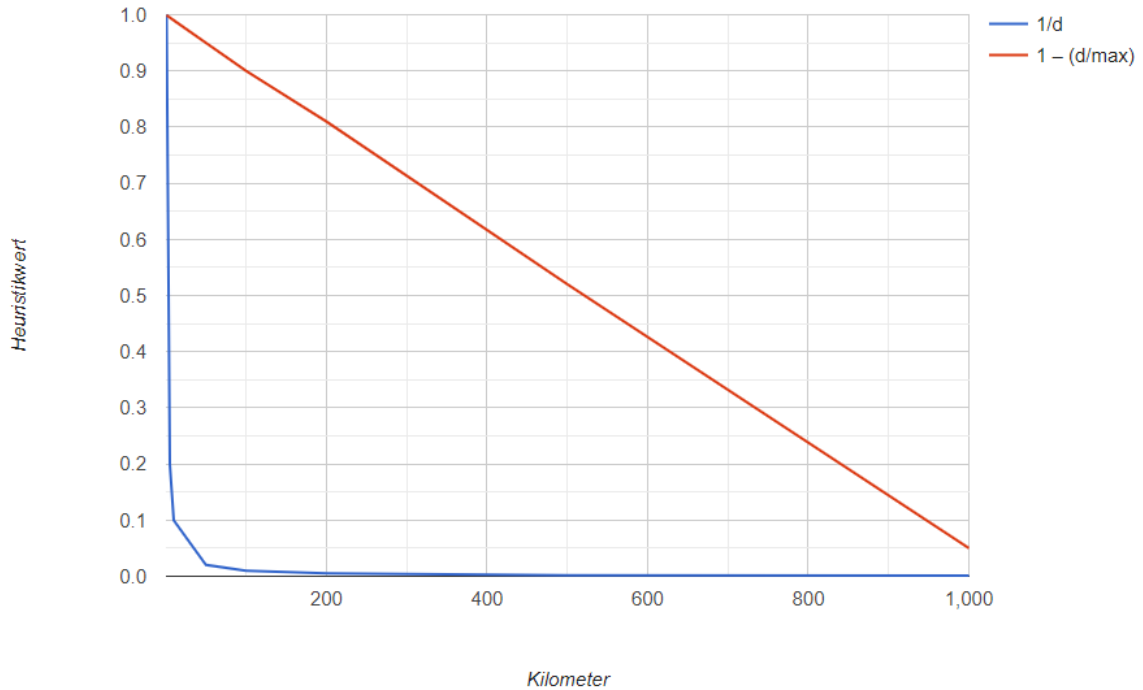


## ALG 3

- Bestimmung Zuordnung zum Bearbeiter
- Reihenfolge je Bearbeiter
  - Globale Matrix für Reihenfolge



# HEURISTIK



## PARAMETER

- Anzahl der Ameisen ( $K$ )
- Anzahl Iterationen ( $N_{Cmax}$ )
- Initialer Pheromon-Wert  $\tau_0$
- Pheromon Verdunstung ( $\rho$ )
  - $\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij}$
- Pheromon Update  $\Delta\tau_{ij}$ 
  - $\tau_{ij} = \tau_{ij} + \Delta\tau_{ij}$
  - $\tau_{ij} = (1-\rho) * \tau_{ij} + \rho * \Delta\tau_{ij}$

## PARAMETER

- Pheromon Update  $\Delta\tau_{ij}$

$$\Delta\tau_{ij} = \frac{1}{L_{best}} \quad \text{wenn } (i, j) \in \text{bester Lösung}$$

$$\Delta\tau_{ij} = 0 \quad \text{ansonsten}$$

$$\Delta\tau_{ij}^f = \frac{Q}{Rank_f} \quad \text{wenn } (i, j) \in \text{bester Lösung der Ameise mit Rang } f$$

$$\Delta\tau_{ij}^f = 0 \quad \text{ansonsten}$$

## PARAMETER

- Pheromon Update  $\Delta\tau_{ij}$

$$\Delta\tau_{ij} = \frac{1}{1 + f(S) - f(S_{best})} \quad \text{wenn } (i, j) \in \text{bester Lösung}$$
$$\Delta\tau_{ij} = 0 \quad \text{ansonsten}$$

$$\Delta\tau_{ij} = 1 \quad \text{wenn } (i, j) \in \text{bester Lösung}$$
$$\Delta\tau_{ij} = 0 \quad \text{ansonsten}$$

## PARAMETER

- Einfluss von Pheromon( $\alpha$ ) und Heuristik ( $\beta$ )

$$p_{ij} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{j \in \text{mögliche Knoten}} \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}$$

- $\beta = (1 - \alpha)$
- Übergangsfunktion-Weiche Q

$$\text{MAX}(\tau_{ij}^{\alpha} * \eta_{ij}^{\beta})$$

# MESSUNGEN

## TESTDATEN

- 200 Bearbeitern
  - Stundenlohn zwischen 10 und 20
- 3000 zuzuordnenden Aufträgen
  - Bearbeitungsdauer: 6 - 16h
  - Abgabe zwischen 0 und  $11 \cdot 3000$
  - Verspätungskosten zwischen 100 und 1000
- Koordinaten in Deutschland
- Eigenschaften {A,B,C,D,E,F,Ob1,Ob2}
- 5 verschiedene Problem-Instanzen



## MESSDATEN

- das Ergebnis der Kosten C
- C je einzelner Iteration
- der maximal benötigte Arbeitsspeicher in der Spitze
- das Endergebnis (Mitarbeiter-Auftrags-Zuweisung)
- die Pheromon-Matrizen nach 5%, 10%, 20%, 40%, 60%, 80% und 100% der Iterationen
- ausgeführter Algorithmus und Heuristik
- den Parametern (Anzahl Ameisen, Verdunstung, ...)
- genutzter Testdatensatz

## VERGLEICH

durchschnittliche Abweichung von der Untergrenze (LB Lower Bound)

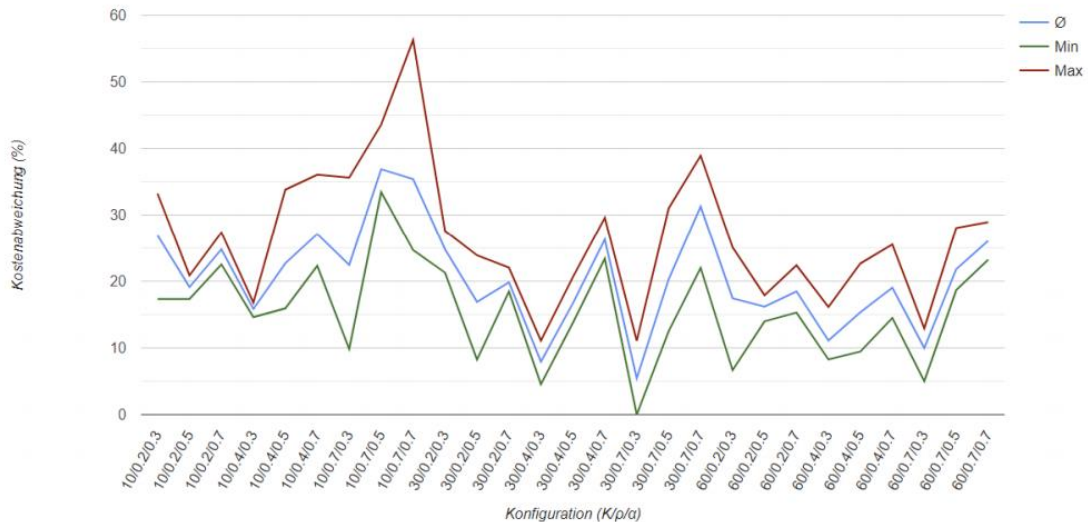
$$\frac{C - LB}{LB} \times 100 \%$$

## REFERENZSET

- Algorithmus: ALG5
- Heuristik:  $1/d$
- Parameter:
  - Anzahl der Ameisen ( $K$ ) = 10
  - Anzahl Iterationen ( $N_{\text{Cmax}}$ ) = 100
  - Pheromon-Verdunstung ( $\rho$ ) = 0.4
  - Einfluss von Pheromon( $\alpha$ ) und Heuristik ( $\beta$ ):  $\beta = 1-\alpha$
  - Übergangsfunktion-Weiche  $Q = 0.1$

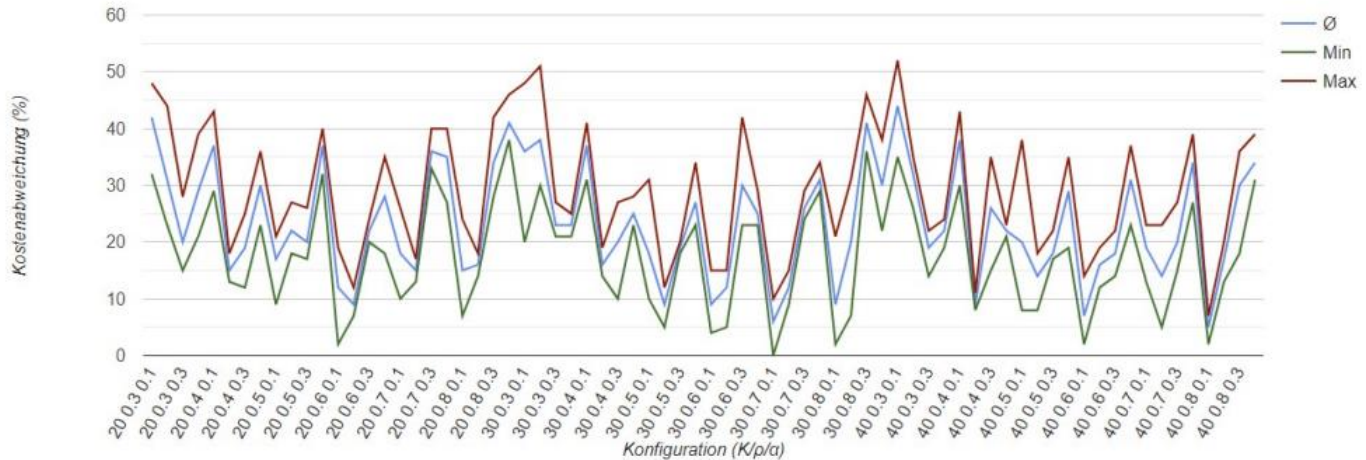
## PARAMETER MESSEN

- K: 10, 30, 60,  $\rho$ : 0.2, 0.4, 0.7,  $\alpha$ : 0.3, 0.5, 0.7
- $3 \times 3 \times 3 = 27$  Messungen (je 10 Mal wiederholt)

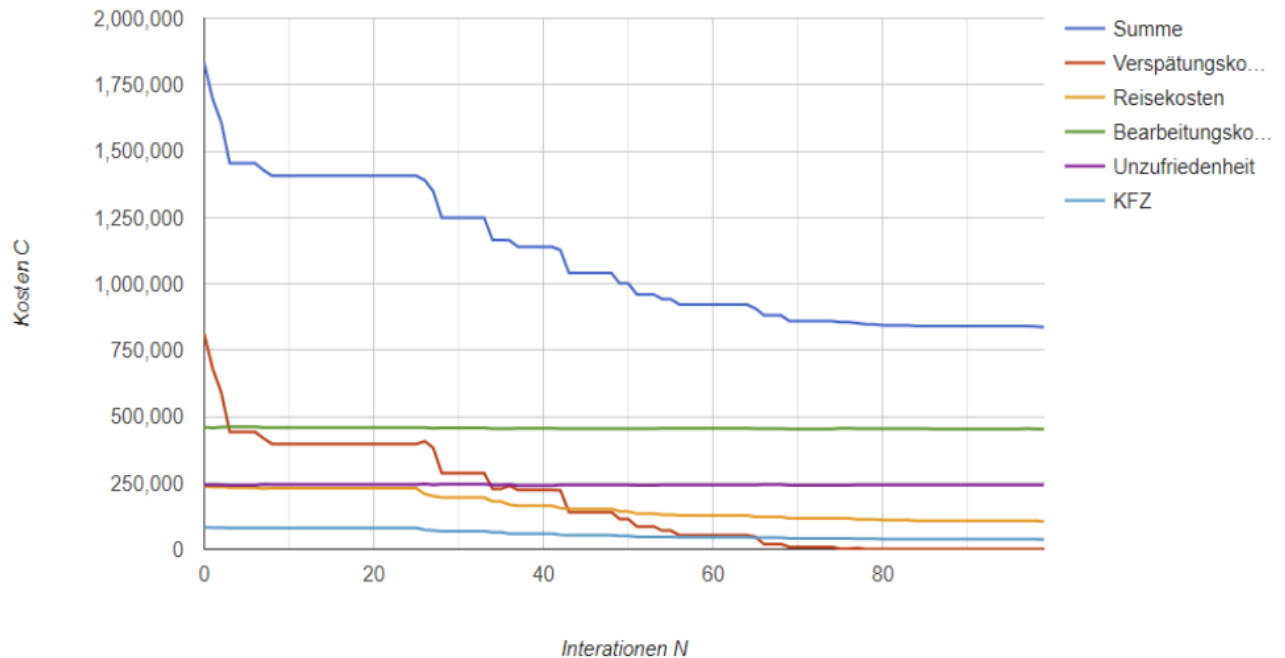


## PARAMETER MESSEN

- K: 20, 30, 40,  $\rho$ : 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,  $\alpha$ : 0.1, 0.2, 0.3, 0.4

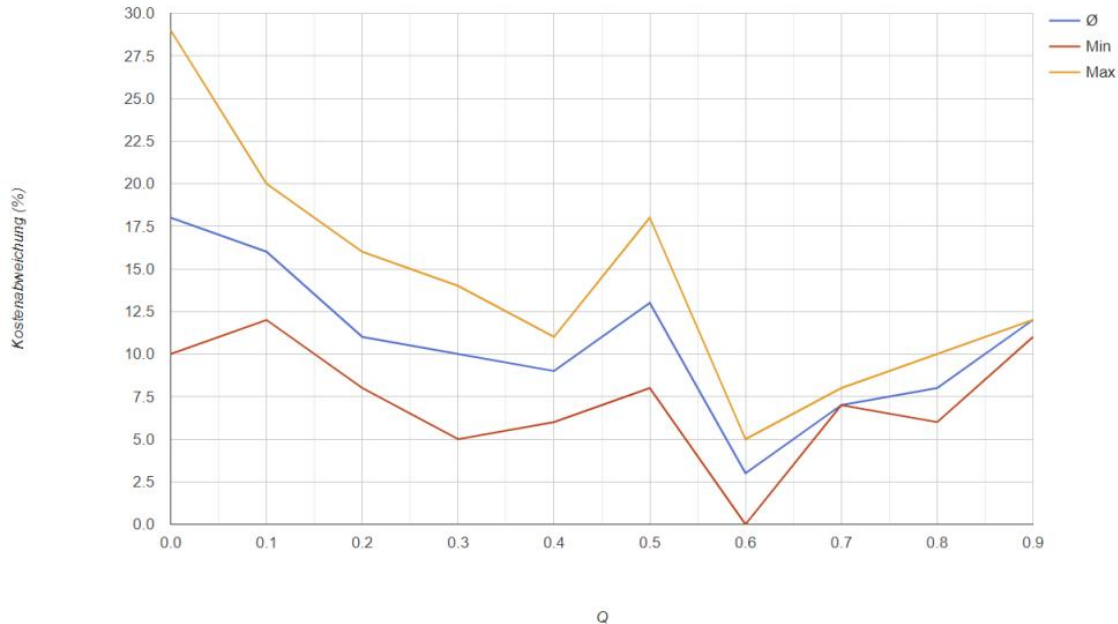


# PARAMETER MESSEN

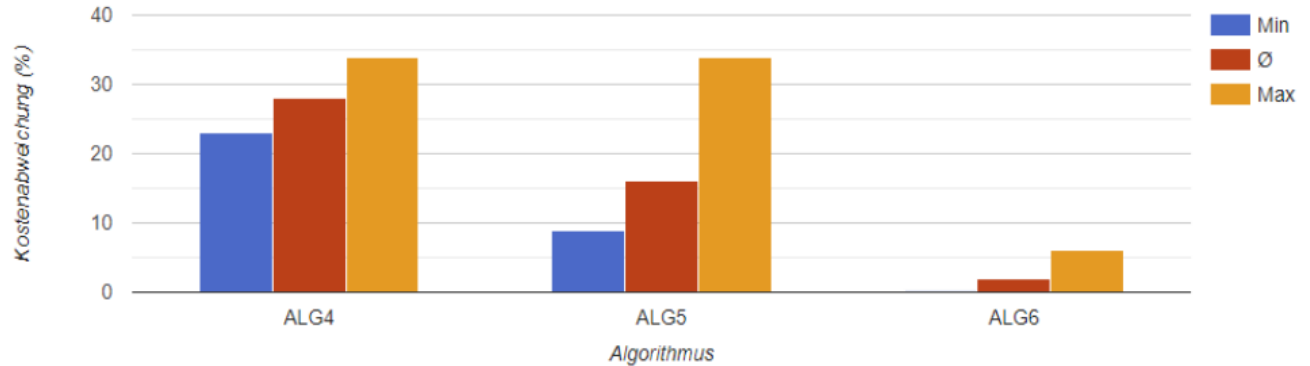


# PARAMETER MESSEN

— Q: 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9

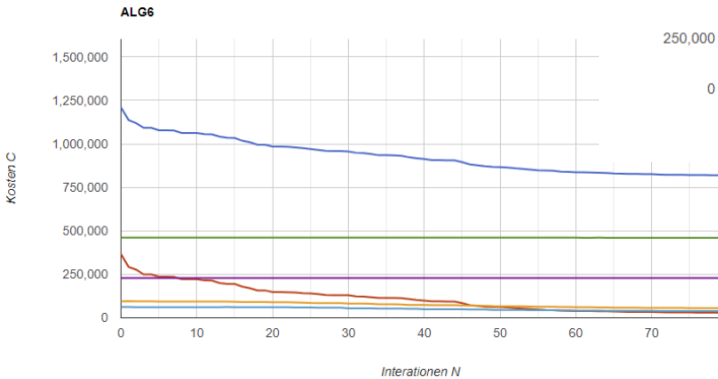
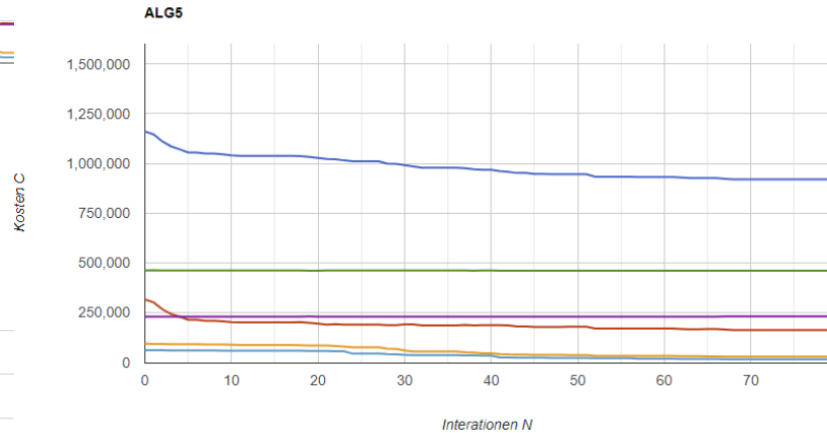
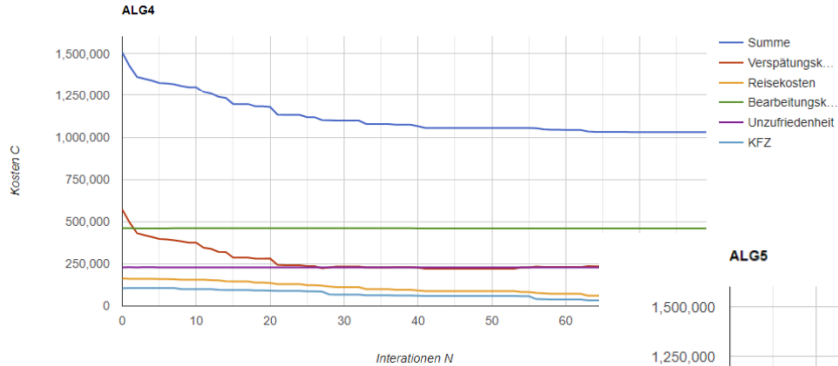


## BESTIMMUNG DES ALGORITHMUS

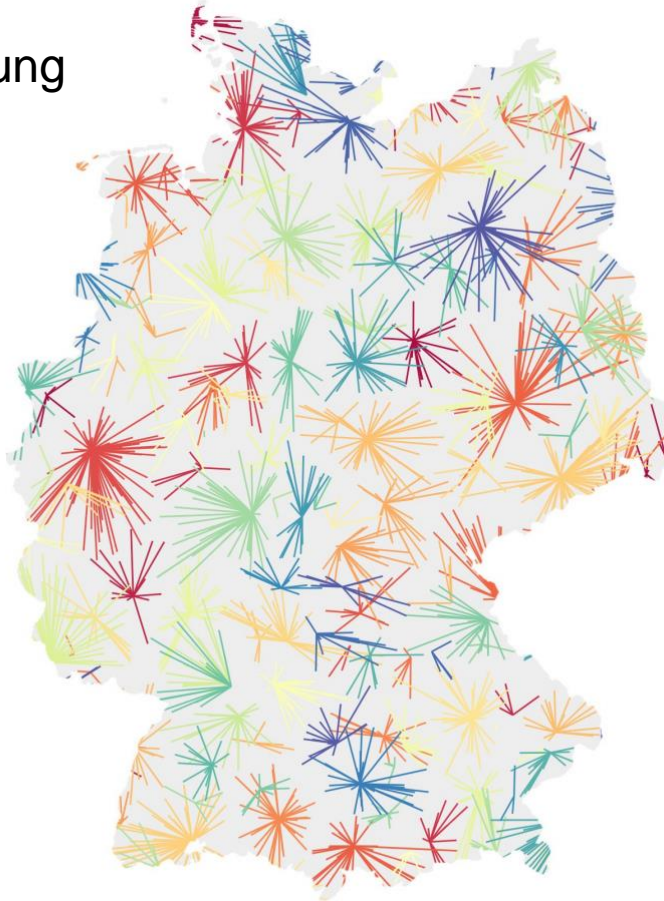




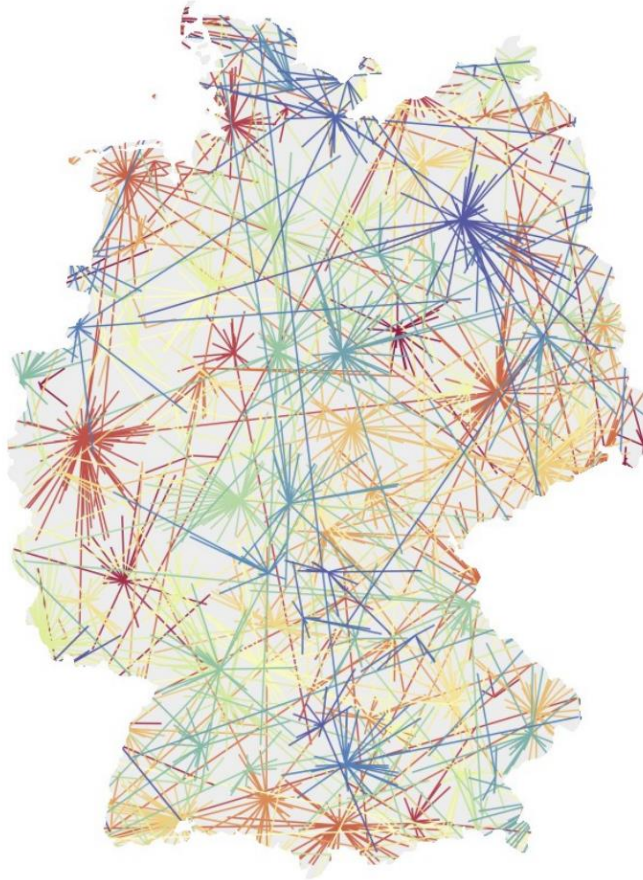
# Auftragsverteilung | 4. Messung



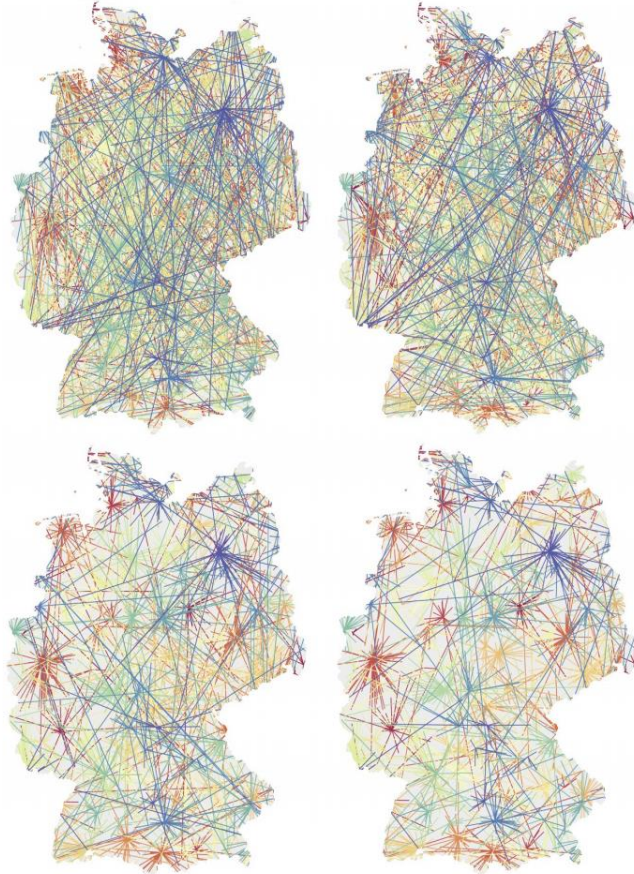
# Auftragsverteilung Greedy-Alg.

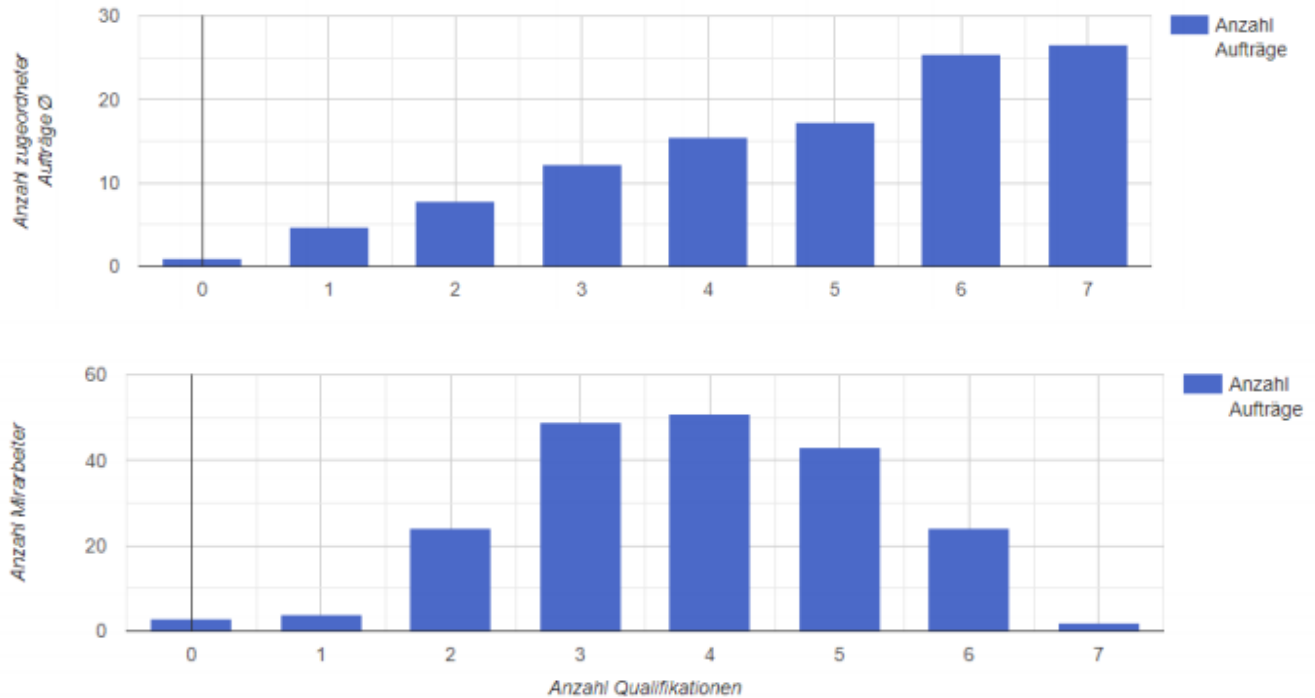


# Auftragsverteilung ALG6



# Auftragsverteilung ALG6 nach 1,20,40,60 Iterationen



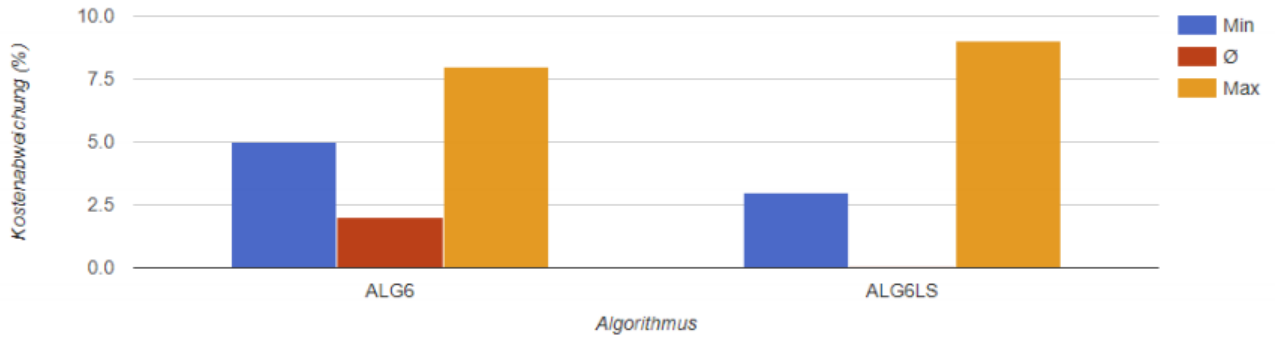


# LOKALE SUCHE

## LOKALE SUCHE

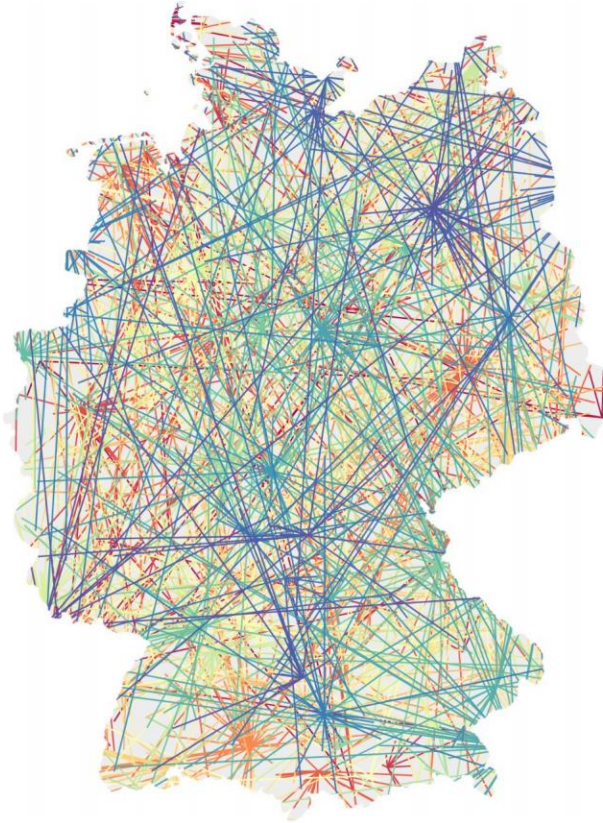
- Das Ergebnis liegt vor
- Generiere für jeden Auftrag eine Zufallszahl  $R \in [0,1]$
- Wenn  $R \leq T$ ,  $T$  ist eine vordefinierte Konstante
  - Für jede Position  $i$  ( $i = 1, \dots, N$ )
  - Platziere den Auftrag an Position  $i$
  - Bestimme die Güte der neuen Lösungen
- $1/T = 10/3000 = 0.3\%$ 
  - Bei 80 Iterationen zusätzliche 2,2 Mio Lösungen

# LOKALE SUCHE

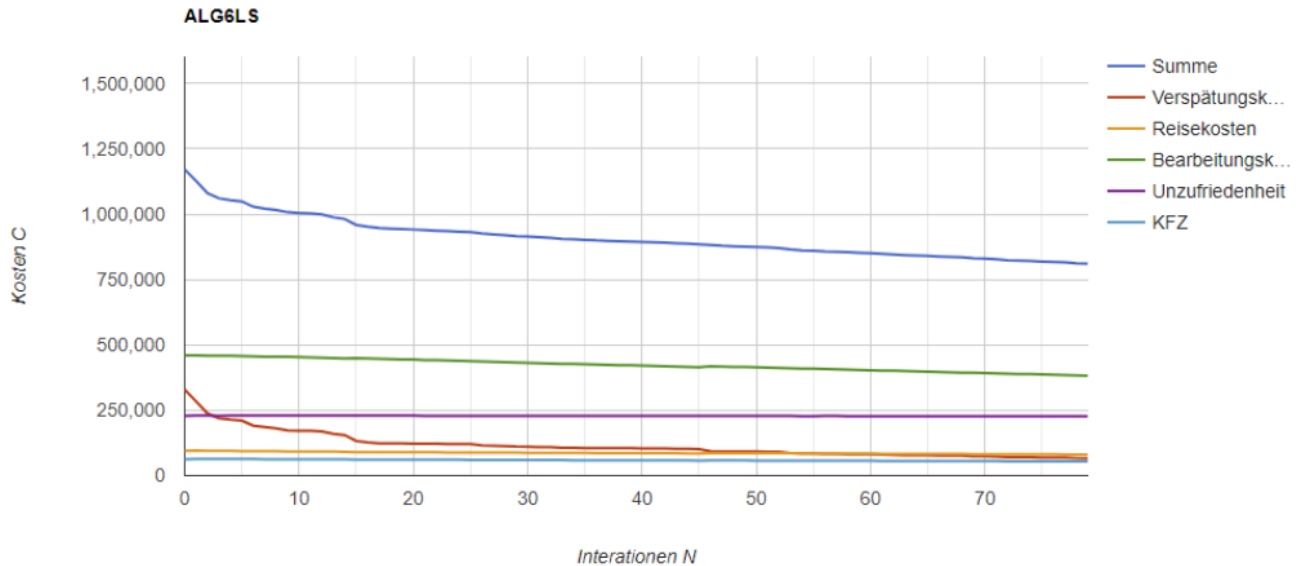




# LOKALE SUCHE



# LOKALE SUCHE



## AUSBLICK

- Verteilung ausgewogener gestalten
  - Anzahl der wenigsten und meisten Aufträge je Bearbeiter nicht zu weit vom Mittel abweicht
  - Die längste Strecken minimieren
- mehrerer Pheromon Matrizen, mehrerer Kolonien und dominanter Fronten zum Update



UNIVERSITÄT  
LEIPZIG

**VIELEN DANK!**