RWTH Aachen University **MAO**
Deutsche Post Lehrstuhl für **WS 2017/18**
Optimierung von Distributionsnetzwerken **19.10.2016**
Prof. Dr. Michael Schneider
Christian Schröder

Übung 1 - Lösungen

Übung 1 - Lösungen19.10.2017
Aufgabe 1 (Allgemeines):

a) Nennen Sie 5 Anwendungen der Optimierung in der Praxis.

- **Demand Planning (Airline):** ***Auf welchen Strecken sollen welche Dienstleistungen/Produkte angeboten werden?***
- **Umlaufplanung (Airline):** ***Welches Flugzeug bedient welches Flugsegment und wird wo/wann gewartet?***
- **Transportplanung (Kurier):** ***Welches Fahrzeug wird wann, wo und wozu eingesetzt?***
- **Standortplanung (Kurier):** ***Wie viele, wie große Depots sollen wo genutzt werden?***
- **Maschinenbelegungsplanung:** ***Welcher Auftrag wird wann auf welcher Maschine bearbeitet?***

b) Grenzen Sie *Simulation* von *Optimierung* ab.
Im Gegensatz zur Optimierung macht die Simulation keinen Entscheidungsvorschlag. Sie dient lediglich dem Vergleich von zuvor ausgewählten Alternativen.

c) Erklären Sie die Bedeutungen der Begriffe *Problem* und *Instanz*?
Das *Problem* bezeichnet einen Satz von Zielfunktion und Nebenbedingungen, welche von konkreten Zahlen abstrahiert sind. Eine *Instanz* ist eine Ausprägung des Problems.

d) Welche Eigenschaften muss eine *optimale* Lösung besitzen?
Die optimale Lösung eines Optimierungsproblems muss *zulässig* sein und den *bestmöglichen* Zielfunktionswert besitzen. Optimale Lösungen müssen *nicht* eindeutig sein!

e) Der Optimierungsprozess ist meistens mit hohen Kosten verbunden. Wie kann man der Frage, ob sich eine Optimierung lohnt, begegnen?

Möglicherweise lässt sich das Optimierungspotential durch eine *untere Schranke* abschätzen. Diese sollte sich möglichst nah am tatsächlichen Optimum befinden und einfach zu berechnen sein.

Aufgabe 2 (Traveling Salesman Problem):

Gegeben sei die in Abb. 1 dargestellte Instanz des Traveling Salesman Problems (TSP). Es wird die Manhattan-Metrik

$$D(\boldsymbol{a}, \boldsymbol{b}) \quad = \quad \sum_i |a_i - b_i|, \quad \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^2 \tag{1}$$

verwendet. Die zugehörige Distanzmatrix ist in Tab. 1 gegeben.

a) Beschreiben Sie Zielfunktion und Nebenbedingungen des TSP.
Gegeben Graph $\mathcal{G}(V, A)$ mit Distanzmatrix d_{ij}.

$$\min z \quad = \quad \sum_{(i,j)\in A} d_{ij} x_{ij} \tag{2}$$

sodass

$$\sum_{(i,j)\in\delta^+(i)} x_{ij} \;=\; 1, \qquad \forall i \in V, \tag{3}$$

$$\sum_{(i,j)\in\delta^-(j)} x_{ij} \;=\; 1, \qquad \forall j \in V, \tag{4}$$

$$\sum_{(i,j)\in A(S)} x_{ij} \;\leq\; |S| - 1, \qquad \forall S \subset V,\ 2 \leq |S| < |V|, \tag{5}$$

$$x_{ij} \;\in\; \{0, 1\}, \qquad \forall (i,j) \in A, \tag{6}$$

dabei bedeutet (3), dass der Ausgangsgrad aller Knoten 1 ist, (4), dass der Eingangsgrad aller Knoten 1 ist, (5), dass es keine Subtouren gibt (zwischen den Knoten S jeder Teilmenge aus V muss eine Kante weniger existieren als Knoten, sie darf also keinen Zyklus enthalten. $A(S)$ bezeichnet alle Kanten zwischen Knoten der Teilmenge S) und (6), dass die Entscheidungsvariablen binär sind.

b) Was ist der Unterschied zwischen dem *symmetrischen* und dem *asymmetrischen* TSP? Um welches handelt es sich im Beispiel?
Beim symmetrischen TSP ist der Graph ungerichtet bzw. die Distanzmatrix symmetrisch. Im asymmetrischen Fall ist der Graph gerichtet, bzw. die Distanzmatrix beliebig. Das vorliegende Beispiel

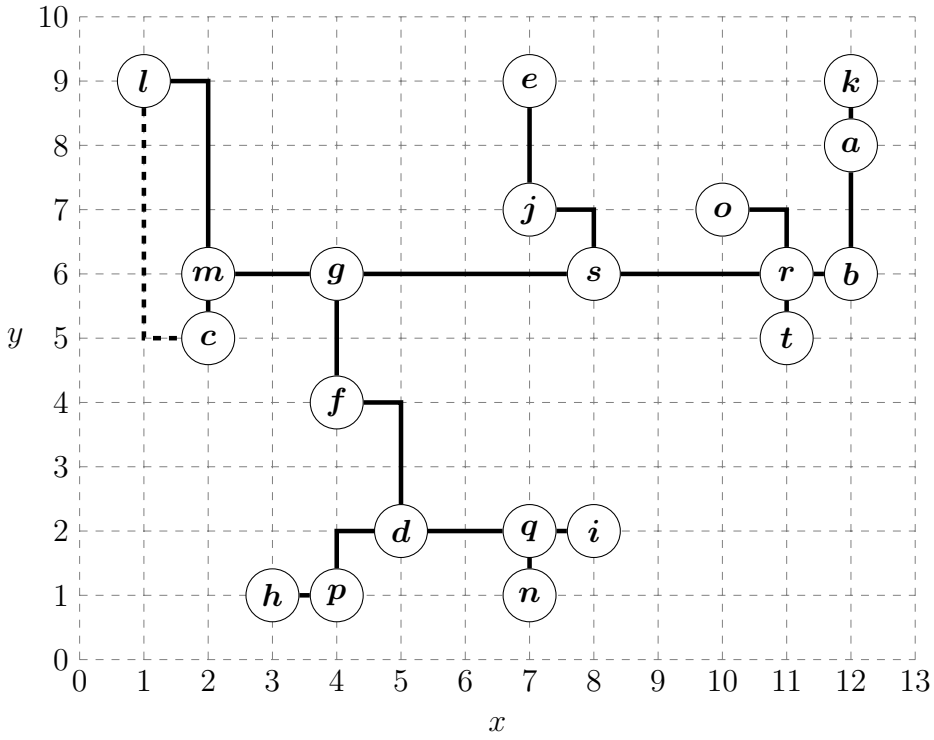

Abbildung 1: TSP Beispielinstanz mit MST und MOT

D	***a***	***b***	***c***	***d***	***e***	***f***	***g***	***h***	***i***	***j***	***k***	***l***	***m***	***n***	***o***	***p***	***q***	***r***	***s***	***t***
a	0	2	13	13	6	12	10	16	10	6	1	12	12	12	3	15	11	3	6	4
b	2	0	11	11	8	10	8	14	8	6	3	14	10	10	3	13	9	1	4	2
c	13	11	0	6	9	3	3	5	9	7	14	5	1	9	10	6	8	10	7	9
d	13	11	6	0	9	3	5	3	3	7	14	11	7	3	10	2	2	10	7	9
e	6	8	9	9	0	8	6	12	8	2	5	6	8	8	5	11	7	7	4	8
f	12	10	3	3	8	0	2	4	6	6	13	8	4	6	9	3	5	9	6	8
g	10	8	3	5	6	2	0	6	8	4	11	6	2	8	7	5	7	7	4	8
h	16	14	5	3	12	4	6	0	6	10	17	10	6	4	13	1	5	13	10	12
i	10	8	9	3	8	6	8	6	0	6	11	14	10	2	7	5	1	7	4	6
j	6	6	7	7	2	6	4	10	6	0	7	8	6	6	3	9	5	5	2	6
k	1	3	14	14	5	13	11	17	11	7	0	11	13	13	4	16	12	4	7	5
l	12	14	5	11	6	8	6	10	14	8	11	0	4	14	11	11	13	13	10	14
m	12	10	1	7	8	4	2	6	10	6	13	4	0	10	9	7	9	9	6	10
n	12	10	9	3	8	6	8	4	2	6	13	14	10	0	9	3	1	9	6	8
o	3	3	10	10	5	9	7	13	7	3	4	11	9	9	0	12	8	2	3	3
p	15	13	6	2	11	3	5	1	5	9	16	11	7	3	12	0	4	12	9	11
q	11	9	8	2	7	5	7	5	1	5	12	13	9	1	8	4	0	8	5	7
r	3	1	10	10	7	9	7	13	7	5	4	13	9	9	2	12	8	0	3	1
s	6	4	7	7	4	6	4	10	4	2	7	10	6	6	3	9	5	3	0	4
t	4	2	9	9	8	8	8	12	6	6	5	14	10	8	3	11	7	1	4	0

Tabelle 1: TSP Distanzmatrix

ist daher symmetrisch.

c) Finden Sie eine möglichst gute untere Schranke für das TSP.

- Relaxieren Sie zunächst die Nebenbedingung an den Knotengrad.
  Eine Zulässige Lösung für das TSP besitzt einen Spannbaum. Der minimale Spannbaum (MST) eines Graphen stellt daher bereits eine untere Schranke dar.
- Versuchen Sie, die so gefundene Schranke zu verbessern.
  Eine einfache Verbesserung dieser Schranke erhält man durch einen sogenannten minimalen 1-Baum (MOT). Einen MOT$_v$ bezüglich des Knotens $v \in V$ erhält man, indem man zunächst einen MST für $V \setminus v$ berechnet und dann v mit seinen beiden kürzesten Kanten an den MST anschließt. Man erhält im Allgemeinen für jeden Knoten $v \in V$ einen unterschiedlichen MOT. Eine TSP Tour lässt sich als Spannbaum über $N-1$ Knoten verstehen, an den ein weiterer Knoten mit zwei Kanten angeschlossen wurde. Daher ist der MOT$_v$ mit dem größten Gesamtgewicht eine untere Schranke für das TSP. Ist der MOT selbst eine TSP Tour, so ist diese optimal.

d) Berechnen Sie die gefundene untere Schranke für die Instanz aus Abb. 1.
Siehe Abb. 1. Für den MST ergibt sich ein Gesamtgewicht von 37. Der MOT mit maximalem Gewicht von 42 wird für den Knoten $\boldsymbol{l}$ erreicht.

e) Der optimale Zielfunktionswert von 50 wird für die Sequenz

$$\boldsymbol{a}, \boldsymbol{k}, \boldsymbol{b}, \boldsymbol{r}, \boldsymbol{t}, \boldsymbol{s}, \boldsymbol{i}, \boldsymbol{q}, \boldsymbol{n}, \boldsymbol{d}, \boldsymbol{p}, \boldsymbol{h}, \boldsymbol{f}, \boldsymbol{g}, \boldsymbol{c}, \boldsymbol{m}, \boldsymbol{l}, \boldsymbol{e}, \boldsymbol{j}, \boldsymbol{o} \tag{7}$$

erreicht. Beurteilen Sie die Qualität der unteren Schranke.
Eine Gute untere Schranke sollte nicht weiter als wenige Prozent unter dem Optimum liegen. Es ergibt sich ein Abstand zur unteren Schranke von

$$\frac{50-42}{50} \cdot 100\% \quad = \quad 16\%, \tag{8}$$

daher ist die untere Schranke von schlechter Qualität. Eine erhebliche Verbesserung dieser unteren Schranke stellt die *Held-Karp* Schranke dar. Dazu sei an dieser Stelle auf die Literatur verwiesen.

Aufgabe 3 (Algorithmik und Komplexitätstheorie):

a) Was ist ein Algorithmus?
Ein *Algorithmus* ist eine genau definierte Verarbeitungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen. Typischerweise wird ein Algorithmus durch eine endliche *Folge von Anweisungen* beschrieben, die nacheinander ausgeführt und oft in festgelegter Weise wiederholt werden.

b) Welche Anforderungen muss ein Algorithmus erfüllen?

- **Eindeutigkeit: die einzelnen Schritte und ihre Abfolge sind unmissverständlich beschrieben.**
- **Allgemeinheit: es wird nicht nur eine Probleminstanz, sondern alle Instanzen eines Problems gelöst.**
- **Ausführbarkeit: der „Prozessor“ muss die Einzelschritte abarbeiten können.**
- **Endlichkeit: seine Beschreibung besteht aus einem Text endlicher Länge.**

c) Was versteht man unter *polynomieller Reduzierbarkeit*?
Das Problem Π_2 heißt *polynomial reduzierbar* auf das Problem Π_1, wenn es polynomial berechenbare Funktionen f und g gibt, wobei

- **f Instanzen von Π_2 in Instanzen von Π_1 transformiert und**
- **g Lösungen von Π_1 in Lösungen von Π_2 transformiert.**

d) Erklären Sie den Unterschied zwischen NP-*Schwere* und NP-*Vollständigkeit* im Rahmen der Komplexitätstheorie.
Ein Problem Π heißt NP-schwer, falls sich jedes Problem $\Pi' \in$ NP polynomial auf das Problem Π reduzieren lässt. Man beachte, dass Π irgendein Problem sein kann, insbesondere muss Π selbst nicht unbedingt zu NP gehören. Ein Problem Π heißt NP-vollständig, falls Π NP-schwer ist und $\Pi \in$ NP gilt.

e) Welchen Einfluss hat die Anzahl möglicher Lösungen für ein Optimierungsproblem auf seine Schwierigkeit?
Die Schwierigkeit eines Problems hängt lediglich von der Komplexitätsklasse ab, in welcher das Problem liegt. Daher hat die größe des Lösungsraums im Allgemeinen keinen Einfluss auf seine Schwierigkeit. Anmerkung: Da $P \neq NP$ lediglich eine Vermutung darstellt, ist die tatsächliche Schwierigkeit der Probleme in NP nicht bekannt.

f) Welche drei Ziele werden beim Entwurf von Algorithmen verfolgt?

- **Allgemeinheit: Der Algorithmus sollte alle Instanzen des Problems Π lösen können.**
- **Effizienz: Der Algorithmus sollte eine polynomiale Laufzeit haben. Anmerkung: Normalerweise nicht höher als dritter Ordnung, da ansonsten auch polynomiale Laufzeitkomplexität schnell unpraktikabel wird. Außerdem bedeutet eine höhere Ordnung nicht immer eine höhere Geschwindigkeit in der Praxis, da bei der Komplexitätsbetrachtung Faktoren und Konstanten vernachlässigt werden. So kann es in der Praxis sein, dass ein Algorithmus mit einer höheren polynomialen Ordnung meistens schneller ist als ein anderer Algorithmus mit niedrigerer Ordnung.**
- **Optimalität: Der Algorithmus sollte exakt sein, d.h. eine optimale Lösung bestimmen.**

Aufgabe 4 (Order Picking):

Eine der Haupttätigkeiten in der Lagerhaltung ist das Kommissionieren von Waren. Das Order Picking Problem besteht darin, den kürzesten Weg durch das Lagerhaus zu finden, so dass alle Positionen von zu kommissionierenden Artikel besucht werden.
Als Beispiel dient das in Abb. 2 abgebildete Regallager. Es besitzt 5 Längsgänge (vertikal). Neben diesem besitzt es einen unteren und oberen sowie einen mittigen Quergang (horizontal). Eine horizontale Reihe von Regalen wird im folgenden als *Block* bezeichnet. Das dargestellte Lager besitzt einen oberen und einen unteren Regalblock. Die zu kommissionierenden Artikel sind durch eingekreiste Buchstaben markiert.
Hinweis: Artikel können nur von einem Längsgang aus entnommen werden.
Eine Möglichkeit, einen Weg durch das Lagerhaus zu planen, bietet die sogenannte *Largest Gap* Strategie:

1. Der Kommissionierer startet am Depot $(1, 0)$ und geht bis zum nächsten Längsgang, in dem sich ein Artikel aus dem Auftrag befindet (über alle Blöcke gesehen). Dann geht er bis zu dem Quergang hinter den obersten Block, in welchem sich zu kommissionierende Artikel befinden.
2. Danach läuft er in jeden Längsgang genau so weit hinein, dass der größte Abstand zwischen zwei aufzunehmenden Artikeln bzw. zum Gangende maximal wird. Das bedeutet, dass er am Längsgang vorbeigeht, wenn der größte Abstand zwischen aktuellem Quergang und dem ersten Artikel liegt.

3. Ist er am letzen Längsgang des Blocks mit Auftragsartikeln angekommen, geht er durch diesen hindurch bis zum nächsten Quergang.
4. Diesen geht er dann bis zum letzten Längsgang des oberen oder unteren Blocks, in dem sich verbleibende Artikel befinden, hindurch. Auf diesem Weg sammelt er alle nicht eingesammelten Artikel aus dem oberen Block ein.
5. Dann verfährt er mit dem nächsten Block nach dem gleichen Prinzip.
6. Ist der Kommissionierer im untersten Quergang angekommen, geht er nach dem letzten aufzunehmenden Artikel zurück zum Depot.

Der Name dieser Strategie kommt daher, dass die nicht gelaufene Strecke in jedem Längsgang maximiert wird, also der *largest gap* zwischen zwei Artikeln bzw. einem Artikel und dem Gangende nicht durchlaufen wird.

a) Planen Sie einen Weg für die Instanz des Order Picking Problems aus Abb. 2 mit Hilfe der Largest Gap Strategie.
Siehe Abb. 2.

b) Um welche Art von Algorithmus handelt es sich bei der beschriebenen Strategie bezüglich der Fixierung von Entscheidungsvariablen?
Die Entscheidungsvariablen werden durch die gelaufenen Kanten repräsentiert. Da diese einmalig nacheinander fixiert werden, handelt es sich um einen *Greedy*-Algorithmus.

c) Ist diese Strategie ein exaktes Verfahren? **Hinweis**: Prüfen Sie, ob sich ein einfaches Gegenbeispiel konstruieren lässt.
Die Strategie ist *kein* exaktes Verfahren. Ein einfaches Gegenbeispiel erhält man, wenn man im Beispiel die Auftragsartikel t, q und c streicht. Würde man in diesem Fall beim ersten Durchlauf des mittleren Querganges auch die Artikel des unteren Blocks einsammeln, erhält man eine kürzere Strecke als mit der Largest Gap Strategie.

d) Welche Laufzeitkomplexität besitzt das Verfahren? Ist es *effizient*?
Die Largest Gap Strategie benötigt eine Iteration über alle Lagerpositionen im Lagerhaus. Das Verfahren hat damit eine lineare Laufzeitkomplexität in abhängigkeit der Anzahl an Lagerpositionen. Im Allgemeinen wird ein Algorithmus mit polynomialer Laufzeitkomplexität als effizient bezeichnet, somit ist die Largest Gap Strategie ein effizienter Algorithmus.

Aufgabe 5 (Metaheuristiken):

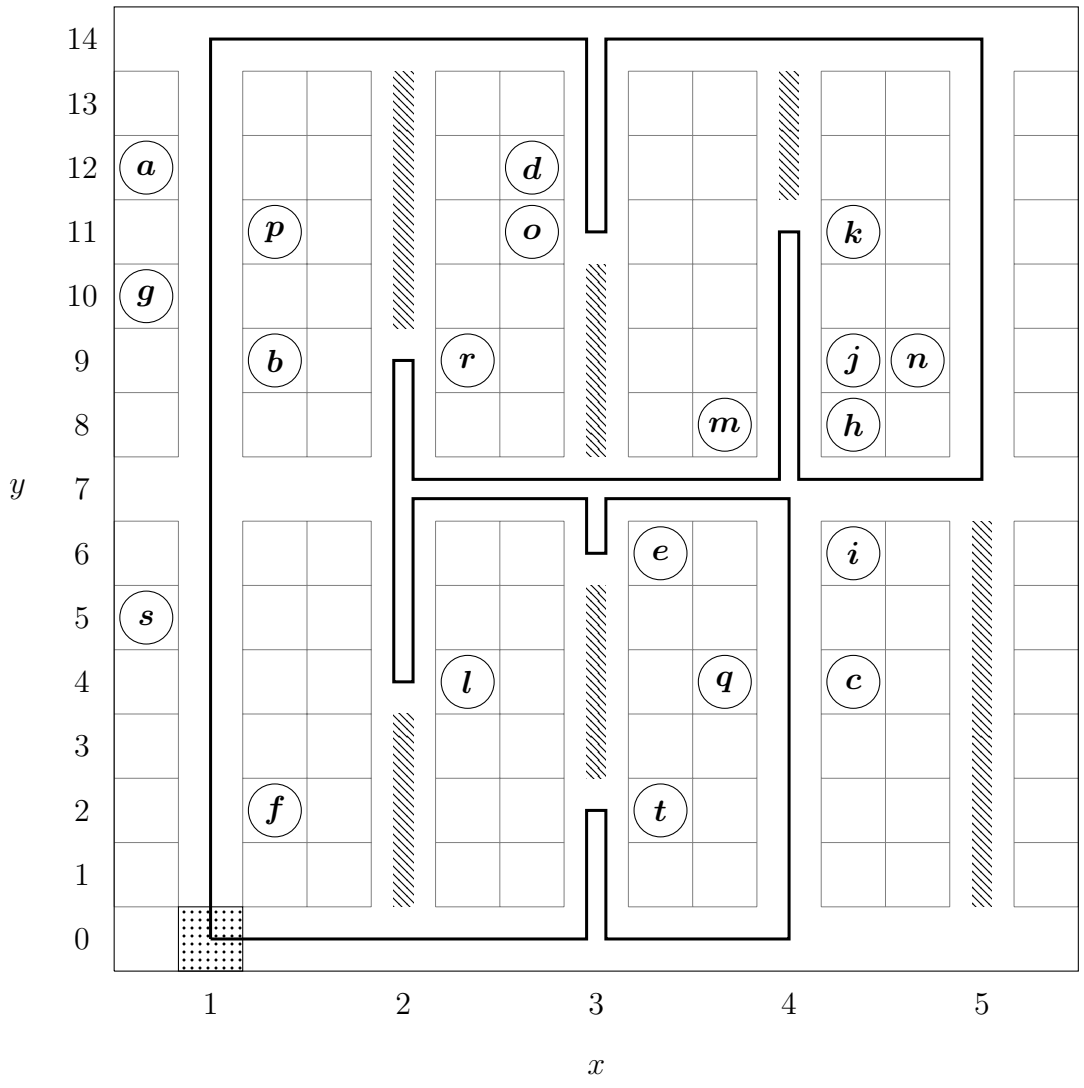

Abbildung 2: Order Picking Beispielinstanz

a) Weshalb werden Metaheuristiken benötigt?
Für viele Probleme lässt sich kein Algorithmus finden, welcher zugleich exakt und effizient ist (P $\neq$ NP Vermutung). Daher ist eine gute Heuristik für viele Probleme eine Vorraussetzung, um praxisrelevante Probleminstanzen zu optimieren. Das liegt daran, dass die in der Praxis vorkommenden Instanzen meistens zu groß sind, um sie mit exakten Algorithmen in vertretbarer Zeit zu lösen.

b) Welche Vor- und Nachteile haben Metaheuristiken?
Zu den Vorteilen gehören die Effizienz und allgemeine Anwendbarkeit auf beliebige kombinatorische Optimierungsprobleme. Der wesentliche Nachteil ist, dass sie im allgemeinen nicht exakt sind.

c) Was unterscheidet eine Metaheuristik von einer Heuristik?
Eine Metaheuristik stellt einen Baukasten von Verfahren dar, welche auf beliebige kombinatorische Optimierungsprobleme anwendbar sind, während Heuristiken im Allgemeinen oft nur auf eine bestimmte Problemklasse anwendbar sind und bestimmte Vorraussetzungen an Instanzparameter haben können.

d) Welche Arten von Metaheuristiken kennen Sie?

- **Simulated Annealing**
- **Tabu-Suche**
- **Ameisen-Algorithmen**
- **Evolutionäre Algorithmen**

e) Handelt es sich bei der Largest Gap Strategie um eine Metaheuristik?
Da die Largest Gap Strategie nur auf Probleminstanzen mit der im Beispiel gegebenen Struktur anwendbar ist, handelt es sich nicht um eine Metaheuristik. Ebenso kann das Verfahren nicht auf beliebige kombinatorische Optimierungsprobleme übertragen werden.

f) Aus welchen Teilen besteht eine Metaheuristik? Zu welchem Teil könnte die Largest Gap Strategie gehören?
Eine Metaheuristik besteht im wesentlich aus zwei Stufen:

1. **Eröffnungsverfahren**
2. **Verbesserungsverfahren**

Die Largest Gap Strategie könnte als Eröffnungsverfahren für Order Picking Probleme zum Einsatz kommen.