

Bin Packing Problem: A general purpose Hill Climbing procedure

Lukas Schmauch, Sebastian Wolf

Seminar Modern Heuristics
Dr. Rico Walter

Februar 2021

Übersicht

Was ist das Bin Packing Problem?

Hill Climbing Ansatz

Computational Studies

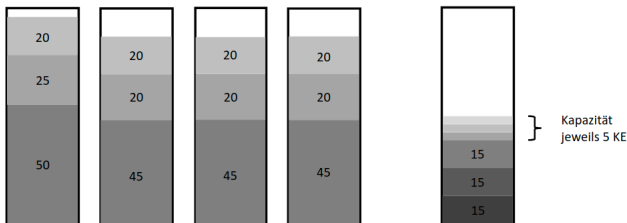
Zusammenfassung

Bin Packing Problem - Kurzübersicht

- ▶ Geg. eine Menge φ mit n unterschiedlichen Items
- ▶ Ziel: Konstruiere die Menge $U = \{U_1, \dots, U_G\}$, wobei:
 - ▶ Alle Items müssen verpackt werden
 $\cup U_i = \varphi$ (for $i = 1, \dots, G$)
 - ▶ Bins enthalten nie das gleiche Item
 $U_i \cap U_j = \emptyset$ (for $1 \leq i \neq j \leq G$)
 - ▶ Es existieren keine leeren Bins
 $U_i \neq \emptyset$ (for $i = 1, \dots, G$)

Grundidee Hill Climbing Ansatz

- Verbesserungsverfahren mit First Fit Ansatz



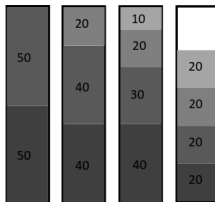
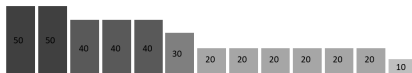
- Lewis, R. (2009): A general-purpose hill-climbing method for order independent minimum grouping problems: A case study in graph colouring and bin packing. Computers Operations Research 36, 2295–2310.

Ablauf Hill Climbing Verfahren

1. Eröffnungsverfahren (First Fit Descending)
2. Zufällige Auswahl von Bins
3. Ausführung des Verbesserungsverfahrens
4. Füge Gruppen wieder zusammen
5. Shuffle der Gruppen
6. Greedy Algorithmus
7. Wiederhole Schritt 2-6 bis Abbruchkriterium erreicht

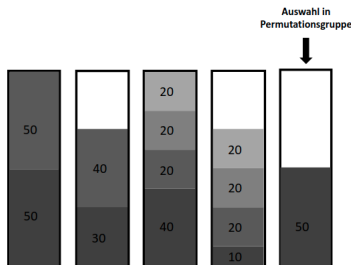
1. Eröffnungsverfahren - First Fit Descending

- ▶ Sortierung nach absteigender Itemkapazität
- ▶ Anwendung des Greedy Algorithmus



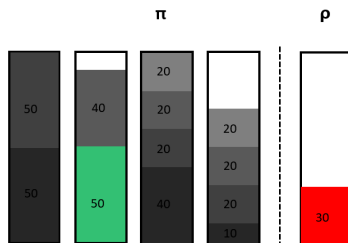
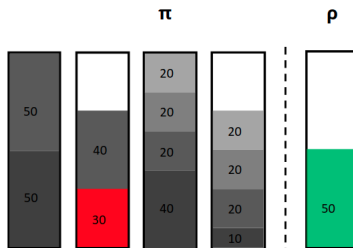
2. Zufällige Auswahl von Bins

- **Standardverfahren:** Auswahl eines Bins mit Wahrscheinlichkeit $p = \frac{1}{\# \text{Gruppen}}$

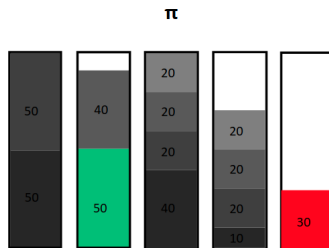


- **Modifikation:** Bin mit minimaler Itemzahl

3. Ausführung des Verbesserungsverfahrens

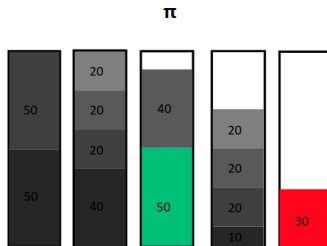


4. Zusammenfügen der Gruppen



5. Shuffle der Gruppen

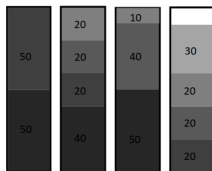
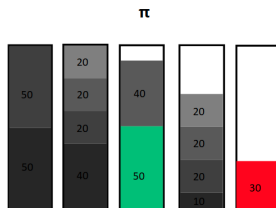
- ▶ **Standardverfahren: 5:5:3 (Largest First, Reverse, Random)**



- ▶ **Modifikation 1:** mittlere Itemkapazität (absteigend)
 - ▶ **Modifikation 2:** inkl. mittlere Itemkapazität (absteigend)
- 5:5:5:3 (Largest First, Reverse, mittlere Itemkapazität, Random)

6. Greedy Algorithmus

- ▶ Ausgangslösung: $|U| = 5$
- ▶ Lösung nach Greedy: $|U'| = 4$
- ▶ **Theorem 1:** $|U'| \leq |U|$



Ablauf Hill Climbing Verfahren

1. Eröffnungsverfahren (First Fit Descending)
2. Zufällige Auswahl von Bins
3. Ausführung des Verbesserungsverfahrens
4. Füge Gruppen wieder zusammen
5. Shuffle der Gruppen
6. Greedy Algorithmus
7. **Wiederhole Schritt 2-6 bis Abbruchkriterium erreicht**

3. Verbesserungsverfahren im Detail

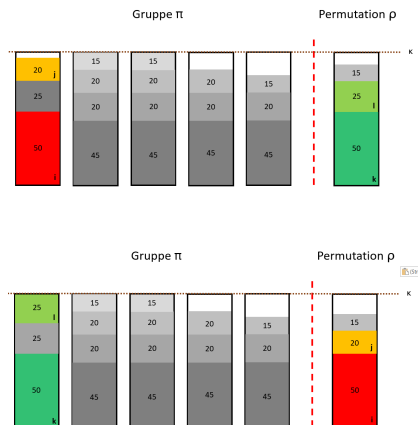
```

1: BPP-IMPROVEMENT-PROCEDURE( $\pi, \rho, C$ )
2: for ( $g = 1$  to  $G(\pi)$ ) do
3:   for each (pair of items  $i, j$  in group  $g$  in  $\pi$ ) do
4:     for ( $h = 1$  to  $G(\rho)$ ) do
5:       for each (pair of items  $k, l$  in group  $h$  in  $\rho$ ) do
6:          $\delta = s[k] + s[l] - s[i] - s[j]$ 
7:         if ( $\delta > 0$  and  $F(g) + \delta \leq C$ ) then
8:           Move items  $i$  and  $j$  into group  $h$  in  $\rho$  and move items  $k$  and  $l$  into group  $g$ 
           in  $\pi$ 
9:   for each (pair of items  $i, j$  in group  $g$  in  $\pi$ ) do
10:    for ( $h = 1$  to  $G(\rho)$ ) do
11:      for each (item  $k$  in group  $h$  in  $\rho$ ) do
12:         $\delta = s[k] - s[i] - s[j]$ 
13:        if ( $\delta > 0$  and  $F(g) + \delta \leq C$ ) then
14:          Move items  $i$  and  $j$  into group  $h$  in  $\rho$  and move items  $k$  into group  $g$  in  $\pi$ 
15:    for each (item  $i$  in group  $g$  in  $\pi$ ) do
16:      for ( $h = 1$  to  $G(\rho)$ ) do
17:        for each (item  $k$  in group  $h$  in  $\rho$ ) do
18:           $\delta = s[k] - s[i]$ 
19:          if ( $\delta > 0$  and  $F(g) + \delta \leq C$ ) then
20:            Move items  $i$  into group  $h$  in  $\rho$  and move item  $k$  into group  $g$  in  $\pi$ 

```

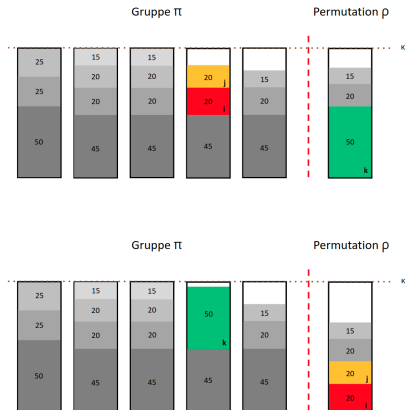
2:2 Move

- ▶ $\delta = s[k] + s[l] - s[i] - s[j]$
- ▶ $\delta = 50 + 25 - 50 - 20 = 5$
- ▶ *if* ($\delta > 0$ and $F(g) + \delta \leq C$)
- ▶ *if* ($5 > 0$ and $95 + 5 \leq 100$)



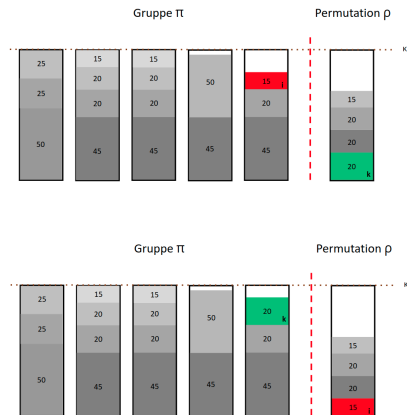
2:1 Move

- ▶ $\delta = s[k] - s[i] - s[j]$
- ▶ $\delta = 50 - 20 - 20 = 10$
- ▶ *if* ($\delta > 0$ and $F(g) + \delta \leq C$)
- ▶ *if* ($10 > 0$ and $85 + 10 \leq 100$)



1:1 Move

- ▶ $\delta = s[k] - s[i]$
- ▶ $\delta = 20 - 15 = 5$
- ▶ *if* ($\delta > 0$ and $F(g) + \delta \leq C$)
- ▶ *if* ($5 > 0$ and $80 + 5 \leq 100$)



Computational Studies - Set-up

Typ	#Instanzen	#Items	Bin-Kapazität	Verteilung	Notation
Uniform	20	120	150	20-100	(U,120,150)
Uniform	20	250	150	20-100	(U,250,150)
Uniform	20	500	150	20-100	(U,500,150)
Uniform	20	1000	150	20-100	(U,1000,150)
Hard	10	200	100000	20000-35000	(H,200,100000)
Triplet	20	60	1000	*	(T,60,1000)
Triplet	20	120	1000	*	(T,120,1000)
Triplet	20	249	1000	*	(T,249,1000)
Triplet	20	501	1000	*	(T,501,1000)

- ▶ * optimale Lösung hat 3 Items pro Bin (ein großes Item und zwei kleine Items)
- ▶ Python 3.8.5
- ▶ Intel Core i7-7Y75, 8GB RAM
- ▶ SEED = 123

Übersicht der Ergebnisse

- ▶ $LB = \left\lceil \sum_{j=1}^n \frac{w_j}{C} \right\rceil$
- ▶ **Uniform:** $LB = z^*$ bei **79 von 80 Instanzen** (eine Instanz mit $z^* = LB + 1$)
- ▶ **Hard:** $LB = z^*$ bei **3 von 10 Instanzen** (Bei 7 von 10 $z^* = LB + 1$)
- ▶ **Triplet:** $LB = z^*$ für **alle Instanzen**

Typ	Inst.	Mittlere LB	FFD	HC	HC*	Mittlere Zeit
(U,120,150)	20	49.1	0.7	0.05	0.1	6.24
(U,250,150)	20	101.6	1.5	0.25	0.25	27.19
(U,500,150)	20	201.2	2.7	0.15	0.15	25.73
(U,1000,150)	20	400.6	4.85	0.25	0.2	42.91
(H,200,100000)	10	55.5/56.2	4.1/3.4	0.8	0	81.04
(T,60,1000)	20	20	3.2	1	0.85	100
(T,120,1000)	20	40	5.8	1	1	100
(T,249,1000)	20	83	12.1	1	1	100
(T,501,1000)	20	167	23.05	1.1	1	100

Optimalitätsanalyse Instanzgruppe Uniform & Hard

► Uniform **20 Instanzen** pro Gruppe und Hard **10 Instanzen**

► $LB = \left\lceil \sum_{j=1}^n \frac{w_j}{C} \right\rceil$

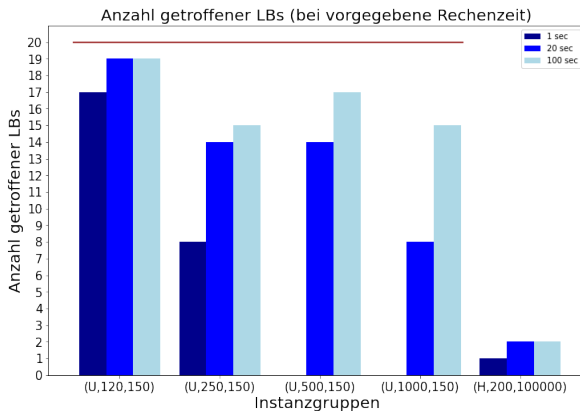


Figure: Anzahl getroffener LBs

Worst Case Analyse Instanzgruppe Uniform & Hard

- ▶ Uniform **20 Instanzen** pro Gruppe und Hard **10 Instanzen**

- ▶ $LB = \left\lceil \sum_{j=1}^n \frac{w_j}{C} \right\rceil$

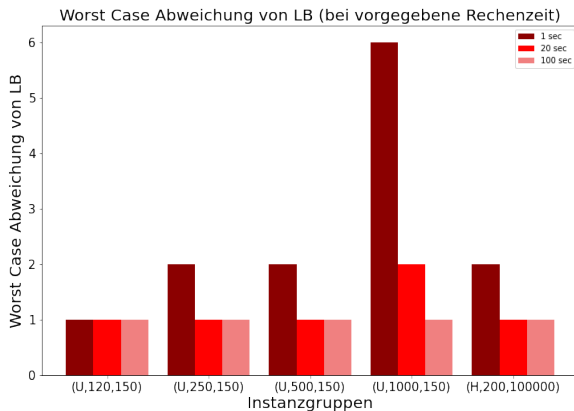


Figure: Worst Case Abweichung von LB

Lösungsgüte im Zeitverlauf Instanzgruppe Uniform & Hard

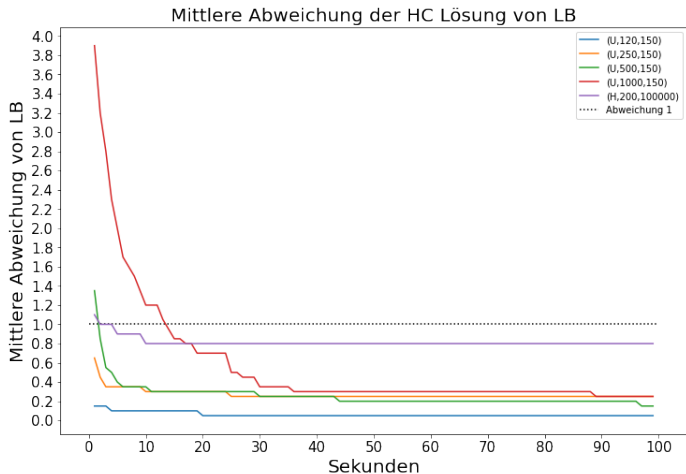


Figure: Mittlere Abweichung von LB pro Zeiteinheit

Optimalitätsanalyse Instanzgruppe Triplet

- ▶ $LB = \left\lceil \sum_{j=1}^n \frac{w_j}{C} \right\rceil = \frac{\#Items}{3}$
- ▶ LB wird nie getroffen

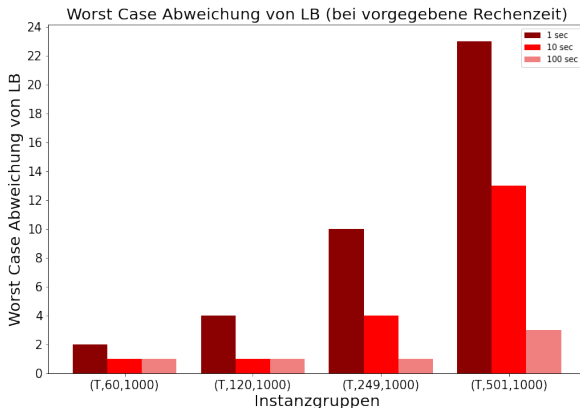


Figure: Worst Case Abweichung von LB

Lösungsgüte im Zeitverlauf Instanzgruppe Triplet

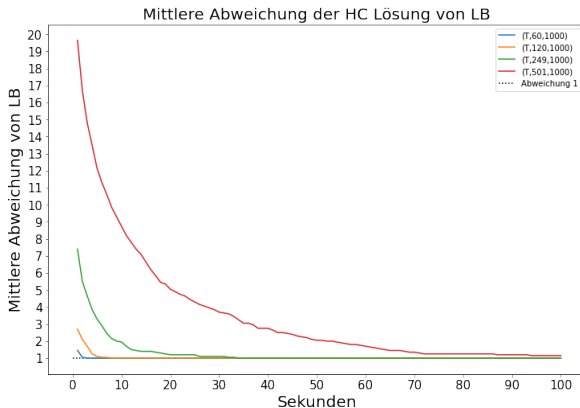


Figure: Mittlere Abweichung von LB pro Zeiteinheit

Vergleich Uniform, Hard und Triplet

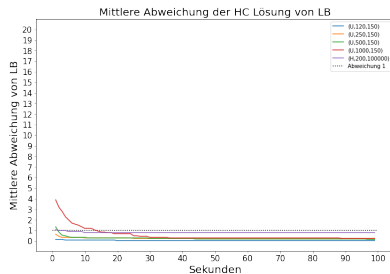


Figure: Uniform und Hard

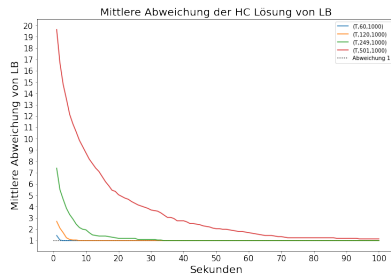


Figure: Triplet

Verteilungsfunktion

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (1)$$

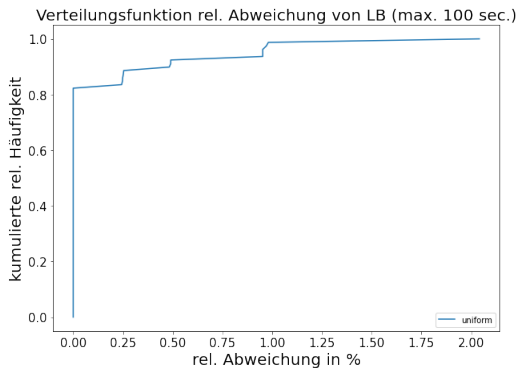


Figure: rel. Abweichung von LB

Verteilungsfunktion

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (2)$$

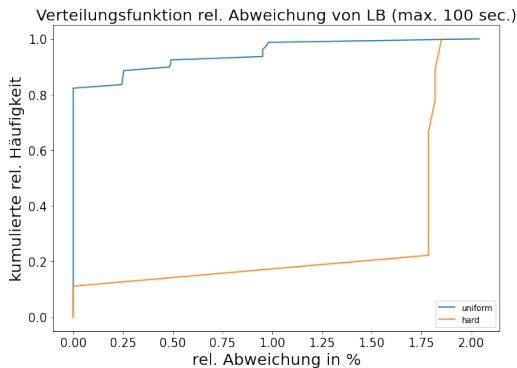


Figure: rel. Abweichung von LB

Verteilungsfunktion

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (3)$$

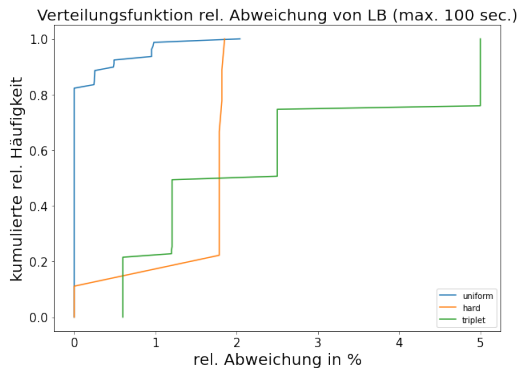


Figure: rel. Abweichung von LB

Verteilungsfunktion - Verfahrensänderung (Uniform)

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (4)$$

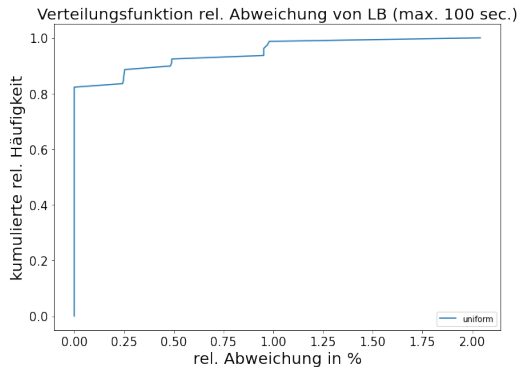


Figure: rel. Abweichung von LB

Verteilungsfunktion - andere Permutationswahl

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (5)$$

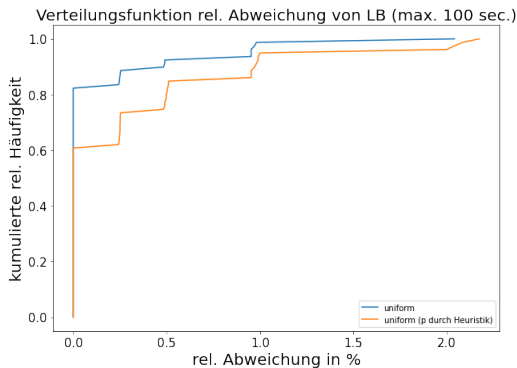


Figure: rel. Abweichung von LB

Verteilungsfunktion - Shuffle (mittlere Itemkapazität)

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (6)$$

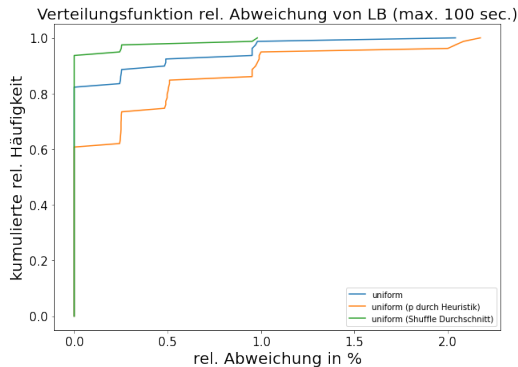


Figure: rel. Abweichung von LB

Verteilungsfunktion - Shuffle (inkl. mittlere Itemkapazität)

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (7)$$

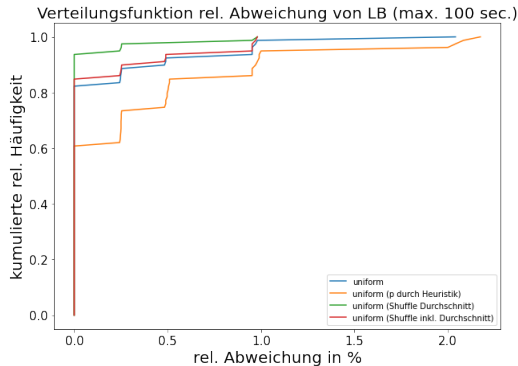


Figure: rel. Abweichung von LB

Verteilungsfunktion - Verfahrensänderung (Triplet)

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (8)$$

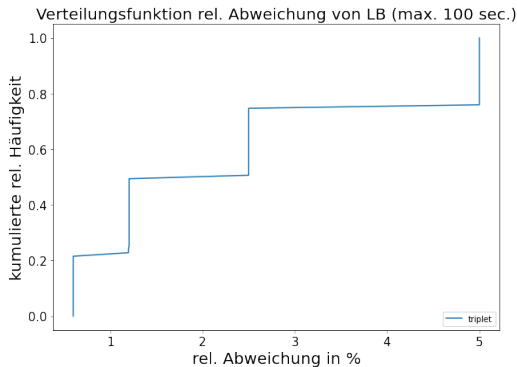


Figure: rel. Abweichung von LB

Verteilungsfunktion - Shuffle (mittlere Itemkapazität)

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (9)$$

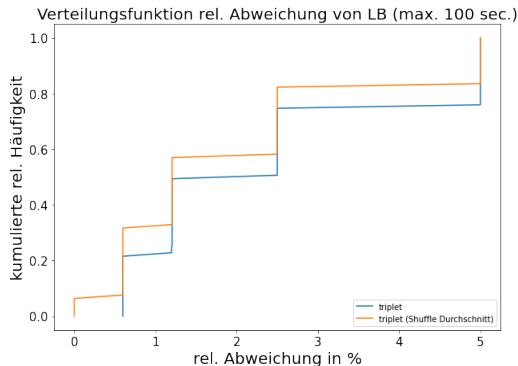


Figure: rel. Abweichung von LB

Verteilungsfunktion - Shuffle (inkl. mittlere Itemkapazität)

$$\text{mit } r = \frac{Z_{HC} - LB}{LB} * 100\% \quad (10)$$

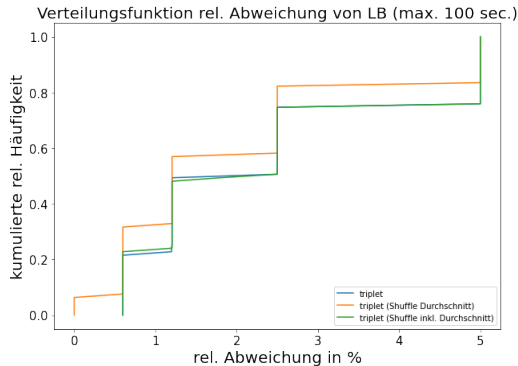


Figure: rel. Abweichung von LB

Zusammenfassung und Ausblick

- ▶ HC-Ansatz zur Lösung des Bin Packing Problems
 - ▶ Eröffnungsheuristik: FFD
 - ▶ Verbesserungsverfahren mit First Fit Ansatz
- ▶ Ausblick Computational Studies
 - ▶ Weitere Untersuchungen des Shuffle-Operators
 - ▶ Weglassen von Moves

Vergleich der Ergebnisse mit weiteren Verfahren

Typ	Mittlere LB	FFD	HC	HC*	IG ¹	HGGA ²	MT ³
(U,120,150)	49.1	0.7	0.05	0.1	0.7	0	0.05
(U,250,150)	101.6	1.5	0.25	0.25	1.45	0	0.55
(U,500,150)	201.2	2.7	0.15	0.15	2.7	0	2.2
(U,1000,150)	400.6	4.85	0.25	0.2	4.85	0	3.85
(H,200,100000)	55.5/56.2	4.1/3.4	0.8	0	2.3	0.1	1.5
(T,60,1000)	20	3.2	1	0.85	2.45	0.6	1.45
(T,120,1000)	40	5.8	1	1	5.3	0.85	4.1
(T,249,1000)	83	12.1	1	1	11.25	0	7.45
(T,501,1000)	167	23.05	1.1	1	22.4	0	14.85

1 Culberson J, Luo F. (1996)

2 Falkenauer E. (1998)

3 Martello S, Toth P. (1990)

Vergleich der Ergebnisse mit weiteren Verfahren

Typ	Mittlere LB	FFD	HC	HC*	HACO ⁴	BISON ⁵
(U,120,150)	49.1	0.7	0.05	0.1	0	-
(U,250,150)	101.6	1.5	0.25	0.25	0.1	-
(U,500,150)	201.2	2.7	0.15	0.15	0	-
(U,1000,150)	400.6	4.85	0.25	0.2	0	-
(H,200,100000)	55.5/56.2	4.1/3.4	0.8	0	-	0.7
(T,60,1000)	20	3.2	1	0.85	-	-
(T,120,1000)	40	5.8	1	1	-	-
(T,249,1000)	83	12.1	1	1	-	-
(T,501,1000)	167	23.05	1.1	1	-	-

4 Levine J, Ducatelle F. (2003)

5 Scholl A, Klein R, Jurgens C. (1997)

Triplet: Vergleich mit Shuffle (mittlere Itemkapazität)

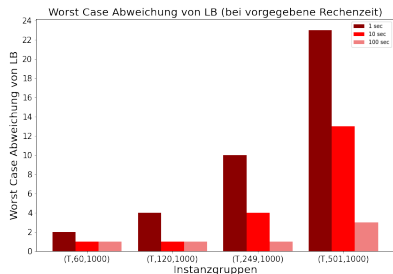


Figure: Standardverfahren

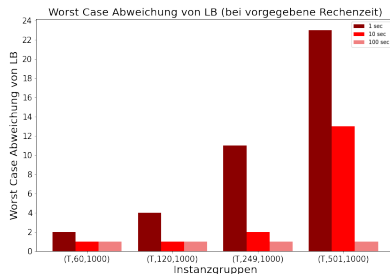


Figure: mittlere Itemkapazität

Uniform: Vergleich mit anderer Permutationswahl

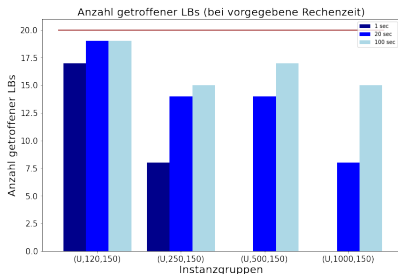


Figure: Random Permutation

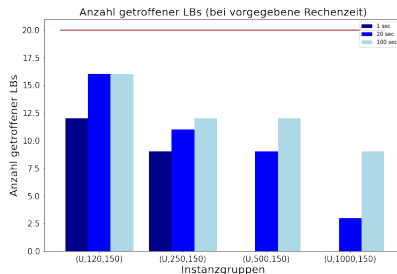


Figure: Minimale Itemzahl

Literaturverzeichnis



Lewis, R. (2009): A general-purpose hill-climbing method for order independent minimum grouping problems: A case study in graph colouring and bin packing. *Computers Operations Research* 36, 2295–2310.



Fleszar et al. (2011): Average-weight-controlled bin-oriented heuristics for the one-dimensional bin-packing problem. *European Journal of Operations Research* 210, 176–184.



Culberson J, Luo F. (1996): Exploring the k-colorable landscape with iterated greedy. In: Johnson DS, Trick MA, editors. *Cliques, coloring, and satisfiability: second DIMACS implementation challenge*, vol. 26. Providence, RI: American Mathematical Society, 245–284.



Falkenauer E. (1998): *Genetic algorithms and grouping problems*. New York: Wiley.



Martello S, Toth P. (1990): Lower bounds and reduction procedures for the bin packing problem. *Discrete Applied Mathematics* 28, 59–70.



Levine J, Ducatelle F. (2003): Ant colony optimisation and local search for bin packing and cutting stock problems. *Journal of the Operational Research Society* 55(12)(7), 705–16.



Scholl A, Klein R, Jurgens C. (1997): Bison: a fast hybrid procedure for exactly solving the one-dimensional bin packing problem. *Computers Operations Research* 25(7), 627–45.