Technical University of Denmark

Integrating load-balancing into multi-dimensional bin-packing problems

Trivella, Alessio; Pisinger, David

Publication date:
2016

Document Version
Peer reviewed version

Integrating load-balancing into multi-dimensional bin-packing problems

TECHNICAL UNIVERSITY OF DENMARK

ALESSIO TRIVELLA
DAVID PISINGER

POZNAŃ, 06.07.2016

Contents

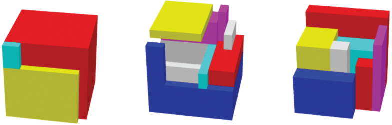
Multi-dimensional Bin-Packing problem (MBP)

Instance

- Set of D-dimensional rectangular-shaped boxes $V = \{1, \ldots, n\}$. Box i has width $= w_{i,d}$ in dimension d
- Identical bins with width $= W_d$ in dimension d

Problem

Orthogonally insert all boxes into the bins avoiding overlapping and using as few bins as possible. Rotations are not allowed



Applications

Shipping and transportation industry, filling up containers, loading trucks etc. Most real-world problems have $D \leq 3$, but all results hold for any dimension

MIP model for the MBP

$$
\begin{aligned}
\min\ & N \\
\text{s.t.}\ & \sum_{d \in D} (l_{ijd} + l_{jid}) + p_{ij} + p_{ji} \geq 1 && \forall\, i < j \in V \\
& x_{id} - x_{jd} + W_d\, l_{ijd} \leq W_d - w_{id} && \forall\ i \neq j \in V,\ d \in D \\
& x_{id} \leq W_d - w_{id} && \forall\ i \in V,\ d \in D \\
& a_i - a_j + n\, p_{ij} \leq n - 1 && \forall\ i \neq j \in V \\
& 1 \leq a_i \leq N && \forall\ i \in V \\
\text{var}:\ & a_j,\ N \in \mathbb{N},\quad x_{id} \in \mathbb{R}^+,\quad l_{ijd},\ p_{ij}\ \in \{0, 1\} && \forall\ i, j \in V,\ d \in D
\end{aligned}
$$

Difficult to solve in practice due to many:

- symmetries
- big-M constraints

Load-Balanced MBP (LB-MBP)

Instance
MBP instance + density of items ρ_i (or mass)

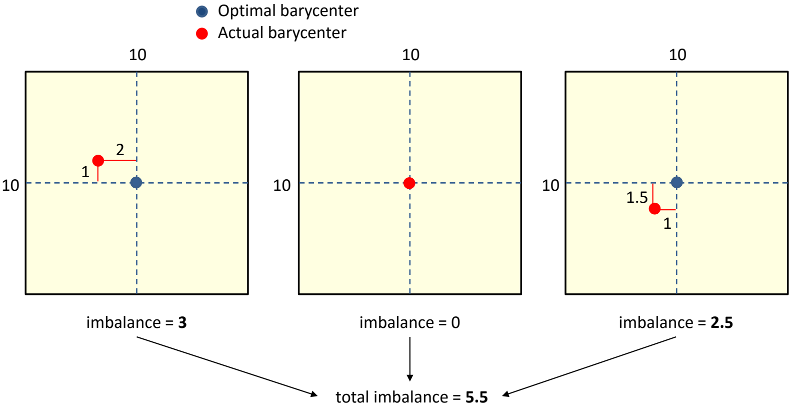
Problem
Arrange items into the miminum number of bins, in such a way that the barycenters of the loaded bins fall as close as possible to an ideal point (e.g. the center of the bin or center of its base)

Applications
Transport (ship, truck, aircraft's cargo): a good position of the center of mass increases the safety and effciency of the travel, minimizing the waste of fuel

Objective function

Minimize the total imbalance over:

- used bins
- dimensions



Balancing a single bin

Assume to have a set V of items which fit into a single bin:

$$\begin{aligned}
\min \quad & \sum_{d \in D} k_d \left(r_d + s_d\right) \\
\text{s.t. :}\ & r_d - s_d = B_d^{opt} - \frac{1}{M}\left(\sum_i m_i \left(x_{id} + \frac{w_{id}}{2}\right)\right) && \forall\, d \in D \\
& \sum_{d \in D} \left(l_{ijd} + l_{jid}\right) \geq 1 && \forall\, i < j \in V \\
& x_{id} - x_{jd} + W_d\, l_{ijd} \leq W_d - w_{id} && \forall\, i \neq j \in V,\ d \in D \\
& x_{id} \leq W_d - w_{id} && \forall\, i \in V,\ d \in D \\
\text{var :}\ & x_{id},\ r_d,\ s_d \in \mathbb{R}^+ \qquad l_{ijd} \in \{0, 1\} && \forall\ i, j \in V,\ d \in D
\end{aligned}$$

To solve the LB-MBP we could:

1) Find the smallest number of bins
2) Balance each bin to optimality

...but the packing and balancing phases are not linked together!

MIP model for the LB-MBP

$$\min \quad N\,C + \sum_{d=1}^{D}\sum_{j=1}^{N} K_d \left(r_{jd} + s_{jd}\right)$$

$$
\begin{aligned}
\text{s.t. :} \quad & \sum_{d=1}^{D} \left(l_{ijd} + l_{jid}\right) + p_{ij} + p_{ji} \geq 1 && \forall\, i < j\\
& x_{id} - x_{jd} + W_d\, l_{ijd} \leq W_d - w_{id} && \forall\, i, j, \forall\, d\\
& a_i - a_j + n\, p_{ij} \leq n - 1 && \forall\, i,\ j\\
& x_{id} \leq W_d - w_{id} && \forall\, i\\
& 1 \leq a_i \leq N && \forall\, i\\
& n\,(c_{ij} - 1) \leq a_i - j \leq n\,(1 - c_{ij}) && \forall\, i,\ j\\
& 1 - (n+1)\,(1 - \delta_{ij}) \leq a_i - j \leq -1 + (n+1)\,(1 - \gamma_{ij}) && \forall\, i,\ j\\
& c_{ij} + \gamma_{ij} + \delta_{ij} = 1 && \forall\, i,\ j\\
& m_i\, W_d \left(c_{ij} - 1\right) \leq e_{ijd} - m_i\, (x_{id} + w_{id}/2) \leq m_i\, W_d \left(1 - c_{ij}\right) && \forall\, i,\ j, \forall\, d\\
& m_i\, W_d \left(c_{ij} - 1\right) \leq \alpha^{ijd} - m_i\, \left(W_d^{opt} - r_{jd} + s_{jd}\right) \leq m_i\, W_d \left(1 - c_{ij}\right) && \forall\, i,\ j, \forall\, d\\
& e_{ijd} \leq c_{ij}\, W_d\, m_i && \forall\, i,\ j, \forall\, d\\
& \alpha_{ijd} \leq c_{ij}\, W_d\, m_i && \forall\, i,\ j, \forall\, d\\
& \sum_{i=1}^{N} e_{ijd} = \sum_{i=1}^{N} \alpha_{ijd} && \forall\, j, \forall\, d
\end{aligned}
$$

$$\text{var :} \quad a_j,\ N \in \mathbb{N}, \qquad x_{id},\ r_{jd},\ s_{jd},\ e_{ijd},\ \alpha_{ijd} \in \mathbb{R}^+, \qquad l_{ijd},\ p_{ij},\ c_{ij},\ \gamma_{ij},\ \delta_{ij} \in \{0, 1\}$$

Sequential vs. joint problem

3D instance with 18 items, $\rho_i = 1$, $B^{opt} = (5, 5, 0)$

Sequential problem

bin 1	1	2	3	4	5	7	8	10
bin 2	6	11						
bin 3	9	12	14	17				
bin 4	13	15	16	18				

Optimal 3DBPP: uses 4 bins

bin	B_x	B_y	B_z	f_{bin}
bin 1	5.00	5.00	4.18	4.18
bin 2	6.12	5.00	4.44	5.56
bin 3	5.00	5.00	4.38	4.38
bin 4	5.00	5.00	3.32	3.32
f_{coord}	1.12	0.00	16.32	**17.44**

Joint problem

bin 1	1	2	3	5		
bin 2	4	6	7	8	10	12
bin 3	9	11	14	17		
bin 4	13	15	16	18		

Different 3DBPP solution

bin	B_x	B_y	B_z	f_{bin}
bin 1	5.00	5.00	4.22	4.22
bin 2	5.00	5.00	3.64	3.64
bin 3	5.00	5.00	4.59	4.59
bin 4	5.00	5.00	3.32	3.32
f_{coord}	0.00	0.00	15.77	**15.77**

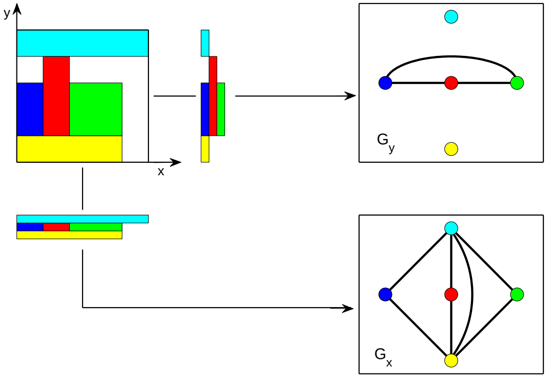
10% improvement!

But running time is 4 vs. 132 seconds. In general the joint model cannot be solved for instances larger than 15-20 items

Heuristic load balancing

We now develop a heuristic algorithm to solve large instances

It is possible to characterize feasible packings by means of a set of Interval Graphs (Fekete-Schepers)



Properties (1/2)

Theorem 1

If D graphs G_d, $d \in D$, are obtained from a packing, then the following conditions are fulfilled:

P_1 : *Each G_d is an interval graph*

P_2 : $\cap_d G_d = \varnothing$

P_3 : *The stable sets of G_d have total weight less than the d-dimension of the bin*

Definition

Let G be an undirected graph. An orientation Φ of G is called transitive orientation (TRO) if:

$$(a, b) \in \Phi \wedge (b, c) \in \Phi \Longrightarrow (a, c) \in \Phi$$

Properties (2/2)

Theorem 2

If G is an interval graph, then its complement $\overline{G}$ is transitively orientable

Theorem 3

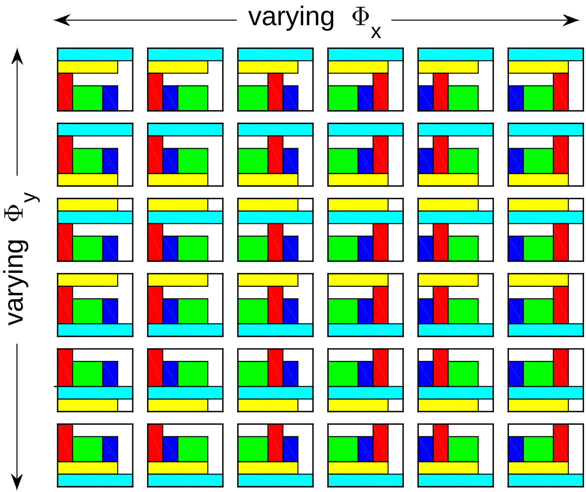
Let G_d, $d \in D$ be D graphs satisfying P_1, P_2, P_3, and call $\Phi = (\Phi_d)_{d \in D}$, where Φ_d is a transitive orientation of $\overline{G}_d$.

The function $p^\Phi : V \longrightarrow {\mathbb{R}_0^+}^D$ defined by:

$$p_d^\Phi(v) = \begin{cases} 0 & \text{if } \nexists\, u \in V : (u, v) \in \Phi_d \\ \max\{\, p_d^\Phi(u) + w_d(u) \,|\, (u, v) \in \Phi_d \,\} & \text{otherwise} \end{cases}$$

produces a packing

How many transitive orientations?



Different transitive orientations produce different packings

Local search among TROs

How many TROs?

From graph theory: number of TROs of a graph is $\prod_{i=1}^{k} r_i!$ where r_i is the number of vertices of particular substructures

Is it possible to find TROs?

From graph theory: we can characterize them all TROs of a graph (it's complicated though)

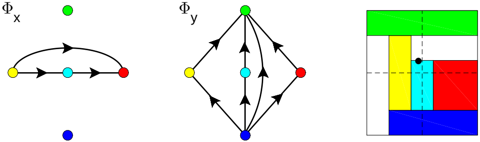
Local Search

We define a best-improvement local search exploring a quadratic neighborhood of TROs. For each TRO:

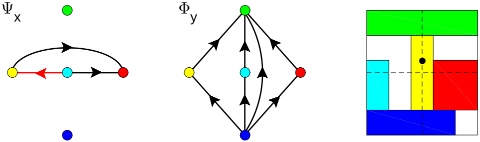
- go back to the corresponding packing
- evaluate the load balancing

Example for a 2D case

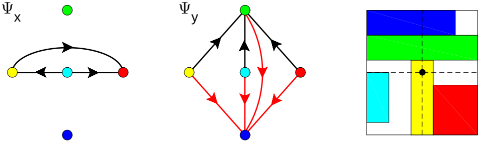
Items have different densities, bin$= 5x5$, $B_{opt} = (2.5, 2.5)$



B = (2.32, 2.97)



B = (2.50, 2.97)



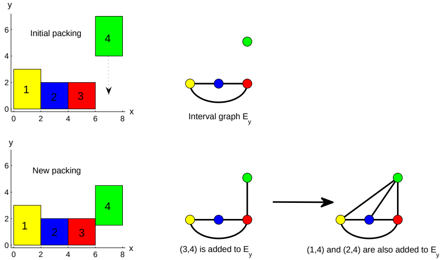
B = (2.50, 2.50)

Local search at graph level

Purpose: improve cases where the number of TROs is limited

How: modifying the structure itself of the graphs:

- Consider interval graphs G_d
- Add or remove edges using specific rules (Crainic et al.)



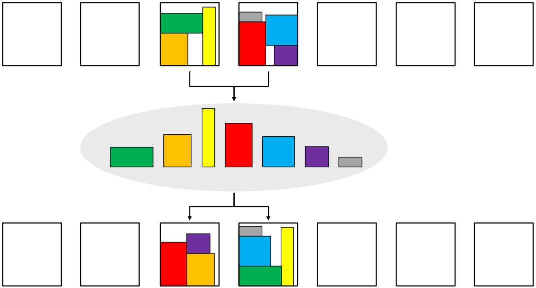
- If new graphs correspond to a packing, then start local search on TROs

Purpose

Exploit the balancing potential of having a large number of bin-packings solutions with the same number of bins

How

Iteratively repack and rebalance n-tuples of weakly balanced bins using *variable-depth neighborhood search* (VDNS)



Local search at bin-packing level (2/2)

Define a *k-neighborhood* as the set of all bin-packing solutions obtained by repacking at most k bins

VDNS algorithm

1. Assign imbalance scores to the bins
2. Select k bins using *roulette wheel selection*
3. Repack the bins using a heuristic for MBP
4. If k bins are still used, balance them
5. If balancing is improved: save solution and update scores

k is dynamically adjusted:
if no solutions are found after n iterations: $k = k + 1$

Results for 3D Bin-Packing instances (1/2)

Optimal barycenter $\left(\frac{W_1}{2}, \frac{W_2}{2}, \frac{W_3}{2}\right)$: center of the bin

				$\rho_i = 1$			$\rho_i \sim U(1,2)$			$\rho_i \sim U(1,6)$		
cl.	size	LB	bins	Init.	Bin	VDNS	Init.	Bin	VDNS	Init.	Bin	VDNS
1	100	24.12	25.64	16.48	0.91	**0.011**	17.03	0.68	**0.054**	19.22	2.26	**0.201**
2	100	24.64	26.12	16.19	0.85	**0.011**	16.69	0.68	**0.042**	19.01	2.28	**0.198**
3	100	24.48	26.08	16.34	0.87	**0.006**	16.78	0.66	**0.030**	19.16	2.21	**0.245**
4	100	57.44	60.60	31.74	0.58	**0.011**	30.64	0.29	**0.030**	30.91	0.93	**0.090**
5	100	13.60	14.60	15.83	0.92	**0.003**	15.87	0.71	**0.068**	17.65	2.70	**0.425**
6	100	18.20	20.08	9.83	1.45	**0.190**	10.30	1.70	**0.396**	13.49	4.54	**1.210**
7	100	11.12	12.36	16.89	1.19	**0.012**	16.53	0.79	**0.050**	18.37	3.11	**0.559**
8	100	15.52	17.08	15.66	0.78	**0.008**	15.34	0.64	**0.056**	17.35	2.63	**0.501**
1	200	48.84	51.16	15.05	0.79	**0.007**	15.74	0.82	**0.032**	18.36	2.57	**0.208**
2	200	48.48	50.80	14.81	0.77	**0.006**	15.57	0.85	**0.044**	18.24	2.58	**0.217**
3	200	49.24	51.24	14.91	0.77	**0.005**	15.68	0.83	**0.036**	18.29	2.65	**0.262**
4	200	117.8	122.2	31.91	0.53	**0.004**	30.66	0.29	**0.012**	30.91	0.96	**0.034**
5	200	25.60	27.36	12.47	0.75	**0.011**	13.06	0.89	**0.147**	14.90	3.23	**0.801**
6	200	35.84	38.24	6.74	1.30	**0.241**	8.17	2.19	**0.523**	12.09	5.44	**1.505**
7	200	20.36	22.40	12.72	1.26	**0.057**	12.95	1.15	**0.329**	14.89	3.99	**1.567**
8	200	29.40	32.04	12.75	0.70	**0.016**	12.79	0.98	**0.183**	14.66	3.33	**0.921**

- Results are average over 25 instances, running time is $<$ 5-10s

Results for 3D Bin-Packing instances (2/2)

Optimal barycenter $\left(\frac{W_1}{2}, \frac{W_2}{2}, 0\right)$: center of the base

				$\rho_i = 1$				$\rho_i \sim U(1,2)$				$\rho_i \sim U(1,6)$			
cl.	size	LB	bins	Init.	Bin	VDNS	LLB	Init.	Bin	VDNS	LLB	Init.	Bin	VDNS	LLB
1	100	24.12	25.64	52.30	43.75	**41.98**	36.06	51.57	42.53	**41.64**	31.87	54.78	41.93	**40.84**	26.82
2	100	24.64	26.12	61.88	46.52	**44.49**	35.90	58.89	43.95	**43.01**	31.72	59.32	40.21	**39.35**	26.68
3	100	24.48	26.08	52.64	43.58	**41.92**	35.56	52.00	42.49	**41.46**	31.42	53.08	41.56	**40.45**	26.41
4	100	57.44	60.60	60.92	39.72	**38.42**	25.59	58.72	38.57	**38.23**	22.61	58.82	38.08	**37.58**	19.06
5	100	13.60	14.60	54.65	44.83	**42.35**	36.77	51.78	42.70	**41.70**	32.44	51.97	41.48	**40.27**	27.16
6	100	18.20	20.08	53.07	47.26	**44.91**	41.98	51.33	45.63	**44.24**	37.07	51.92	44.49	**43.13**	31.13
7	100	11.12	12.36	55.24	44.94	**42.36**	36.59	51.87	42.58	**41.59**	32.30	51.96	41.80	**39.80**	27.12
8	100	15.52	17.08	54.77	44.83	**42.08**	37.08	51.97	43.03	**41.99**	32.72	52.37	42.05	**40.45**	27.44
1	200	48.84	51.16	51.98	44.00	**42.56**	36.87	51.77	43.18	**42.14**	32.58	52.90	42.31	**41.09**	27.43
2	200	48.48	50.80	61.86	47.10	**45.20**	36.79	59.34	44.72	**43.65**	32.51	59.96	40.72	**39.93**	27.36
3	200	49.24	51.24	52.38	44.11	**42.67**	36.68	52.18	43.26	**42.22**	32.14	53.45	42.35	**41.16**	27.28
4	200	117.8	122.2	60.31	39.50	**38.19**	25.74	58.29	38.40	**38.05**	22.76	58.55	37.92	**37.36**	19.18
5	200	25.60	27.36	53.89	45.90	**44.14**	39.29	52.23	44.57	**43.48**	34.71	52.93	43.54	**42.16**	29.21
6	200	35.84	38.24	52.24	48.28	**45.77**	44.36	51.06	46.74	**45.12**	39.19	51.83	45.34	**43.64**	32.97
7	200	20.36	22.40	53.96	46.47	**44.51**	39.88	52.31	44.87	**43.90**	35.23	53.43	44.04	**42.72**	29.64
8	200	29.40	32.04	53.68	45.84	**44.37**	39.61	52.14	44.73	**43.72**	34.98	53.09	45.84	**42.40**	29.42

- Results are average over 25 instances, running time is $<$ 5-10s
- LLB is a lower bound obtained from “liquifying” the items

Background paper

Alessio Trivella and David Pisinger. “The load-balanced multi-dimensional bin-packing problem”. *Computers & Operations Research* 74 (2016)152–164

Thank you.