Can Sağtürk
C.Sagturk@student.tudelft.nl

An Integer Programming Approach to the Multi-Level Bin Packing Problem with Partial Orders

Matthias Horn
Neil Yorke-Smith

1 Background

MLBPPO: MLBP with partial orders. If item A has precedence over item B, then the index of A's top-level bin needs to be higher than or equal to B's bin.

IP: An NP-Hard problem containing a set of integer variables and linear (in)equalities, where the aim is to minimize / maximize a linear objective function.

2 Research Question

"Which IP models can solve the MLBP and MLBPPO problems with an optimal result in reasonable time on small to medium problem instances?"

3 Method

1. Formulate two IP models for each problem, two for MLBP and two for MLBPPO
2. Implement the models in CPLEX
3. Test models on problem instances of varying sizes, measure speed and Branch and Bound nodes
4. Interpret the results

IBM ILOG CPLEX: An optimizer developed to solve linear programming problems.

[1] B. Gavish and S. C. Graves, "The travelling salesman problem and related problems", Working paper GR-078-78, 1978.

[2] R. T. Wong, "Integer programming formulations of the traveling salesman problem", in Proceedings of the IEEE international conference of circuits and computers, IEEE Press Piscataway NJ, 1980, pp. 149–152.

4 MLBP Model 1

The baseline model. Adaptation from an IP model for the bin packing problem.

$$\min \sum_{i=1}^{m}\sum_{j=1}^{n_i} y_j^i \, c_j^i \qquad (1)$$

$$\sum_{k=1}^{n_1} x_{j,k}^0 = 1 \qquad j \in B_0 \qquad (2)$$

$$\sum_{k=1}^{n_{i+1}} x_{j,k}^i \iff y_j^i \qquad i \in \{1 \dots m-1\}, j \in B_i \qquad (3)$$

$$\sum_{j=1}^{n_{i-1}} s_j^{i-1} \, x_{j,k}^{i-1} \le y_k^i \, w_k^i \qquad i \in \{1 \dots m\}, k \in B_i \qquad (4)$$

5 MLBP Model 2

Extends MLBP model 1, (1)-(4). Applies Single Commodity Flow Formulation[1]. Adds weak constriants for stronger LP relaxations.

$$\sum_{k=1}^{n_1} f_{j,k}^0 = 1 \qquad j \in B_0 \qquad (5)$$

$$\left(\sum_{k=1}^{n_{i-1}} f_{k,j}^{i-1}\right) - \left(\sum_{k=1}^{n_{i+1}} f_{j,k}^{i}\right) = 0 \qquad i \in \{1 \dots m-1\}, j \in B_i \qquad (6)$$

$$\sum_{k=1}^{n_m}\sum_{j=1}^{n_{m-1}} f_{j,k}^{m-1} = n_0 \qquad (7)$$

6 MLBPPO Model 1

Extends MLBP model 1, (1)-(4). Variable p traces items to the top level bins.

$$\sum_{k=1}^{n_{i+1}} p_{i,j}^h = 1 \qquad i \in \{0 \dots m-1\}, h \in B_0 \qquad (8)$$

$$p_{0,k}^j = x_{j,k}^0 \qquad j \in B_0, k \in B_1 \qquad (9)$$

$$(p_{i-1,j}^h \;\&\; x_{j,k}^i) \implies p_{i,k}^h \qquad h \in B_0, i = \{1 \dots m-1\}, j \in B_{i-1}, k \in B_i \qquad (10)$$

$$\sum_{k=1}^{n_d} p_{m-1,k}^{o_{first}} \ge \sum_{k=1}^{n_d} p_{m-1,k}^{o_{second}} \qquad d \in \{0 \dots m\}, o \in O \qquad (11)$$

7 MLBPPO Model 2

Extends MLBP model 1, (1)-(4). Applies Multi Commodity Flow Formulation[2]. Adds strong constraints for stronger LP relaxations and to check feasibility.

$$\sum_{k=1}^{n_1} f_{j,k,j}^0 = 1 \qquad j \in B_0 \qquad (12)$$

$$\left(\sum_{k=1}^{n_{i-1}} f_{k,j,h}^{i-1}\right) - \left(\sum_{k=0}^{n_{i+1}} f_{j,k,h}^{i}\right) = 0 \qquad i = \{1 \dots m-1\}, j \in B_i, h \in B_0 \qquad (13)$$

$$\left(\sum_{k=1}^{n_{m-1}} f_{k,j,h}^{m-1}\right) - f_{j,1,h}^{m} = 0 \qquad j \in B_m, h \in B_0 \qquad (14)$$

$$\sum_{j=1}^{n_m} f_{j,1,h}^m = n_0 \qquad h \in B_0 \qquad (15)$$

$$x_{j,k}^i \ge f_{j,k,h}^i \qquad i = \{0 \dots m-1\}, j \in B_i, k \in B_{i+1}, h \in B_0 \qquad (16)$$

$$\sum_{j=1}^{n_d} f_{j,1,o_{first}}^m \ge \sum_{j=1}^{n_d} f_{j,1,o_{second}}^m \qquad d \in \{0 \dots m\}, o \in O \qquad (17)$$

8 Results Summary and Discussion

- Both MLBP models can solve up to 50-item 5-level instances with minor timeout percentages (maximum 24%, much lower for most instance groups). Most instances above 60 items and 3 levels timeout, though the average optimality gap for these is 2.32%.
- No MLBP model is decisively better, superior results are inconsistent.
- For the MLBPPO, both models struggle with time after 2 levels and 30 items. Over 50% failed results for 40 item instances. More precedence pairs can increase the time required for a solution by a factor of 1.5.
- The first model for MLBPPO is decisively better time-wise than model 2, while model 2 required fewer Branch-and-Bound nodes for almost all cases compared to model 1.

9 Conclusions and Future Research

- Both MLBP models can solve medium instances in a reasonable time, where neither model is definitively superior. The first MLBPPO can do the same only for up to 20 items, and the second model cannot compete.
- Since MLBPPO model two looks promising, A hybrid model, formulated and implemented as part LP part network flow optimizer, warrants future work.