Integer Programming Models for the Class Constrained Multi-Level Bin Packing Problem

A.N. Kordyl

Supervised by: N. Yorke-Smith, M. Horn

Delft University of Technology

1. INTRODUCTION

- The **Multi-Level Bin Packing (MLBP)** problem is a generalization of the widely-known **NP-hard Bin Packing (BP)** problem.
- **Task**: Allocate items to first-level bins while minimizing the cost of bins used. Then, fit first-level bins into next-level bins. Repeat until top level is reached.
- **Class Constrained MLBP (CCMLBP)**: MLBP with added Class Constraints. All items belong to one of q classes, and on each level i there exists a bound Q^i, such that the number of items of one class in each bin is not greater than Q^i.
- Use **Integer Programming (IP)** to model these problems.
- Use **CPLEX** to solve **IP** formulations.

2. METHODOLOGY

1. Generate **IP** two formulations of the **MLBP** and **CCMLBP** problems each.
2. Implement these models in the provided **C++** framework to be solved by **CPLEX**.
3. Run the formulations on a variety of instances on the DelftBlue cluster.
4. Evaluate results w.r.t. CPU time, number of branch-and-bound nodes needed, number of solved instances.
5. Find which performs best and how large instances can be solved.

3. RESEARCH QUESTIONS

Which of the considered IP models of the MLBP and CCMLBP problems perform best?

How far can Integer Programming can be used to solve instances of the MLBP and CCMLBP problems?

How large instances can be solved by CPLEX before having to resort to approximation algorithms?

4. MATHEMATICAL FORMULATIONS

Standard MLBP

Variables: $x_{k,i,j} \in \{0,1\}, y_{k,j} \in \{0,1\}$

Objective Function: $\min \sum_{k=1}^{m} \sum_{j=0}^{n_k} y_{k,j} \cdot c_{k,j}$

Constraints:

- All items inserted into level 1 bins.
- All used bins inserted into next-level bins.
- No bin is filled over capacity.

Network Flow Formulation

Added flow variable: $0 \leq f_{k,i,j} \leq n_0$

Constraints:

- Each item as a node emits one unit of flow.
- Flow leaving each bin node must equal flow entering.
- Sum of flow leaving top-level bin nodes equals number of items.
- Max flow per bin = number of items.

CCMLBP Constraints

Added class variable: "is class r in bin j on level k?" $c_{k,r,j} \in \{0,1\}$

Constraints:

- If a bin/item is inserted into another bin, the class(es) of that item/bin must be transferred.
- On each level, the amount of classes in one bin must be below the given bound

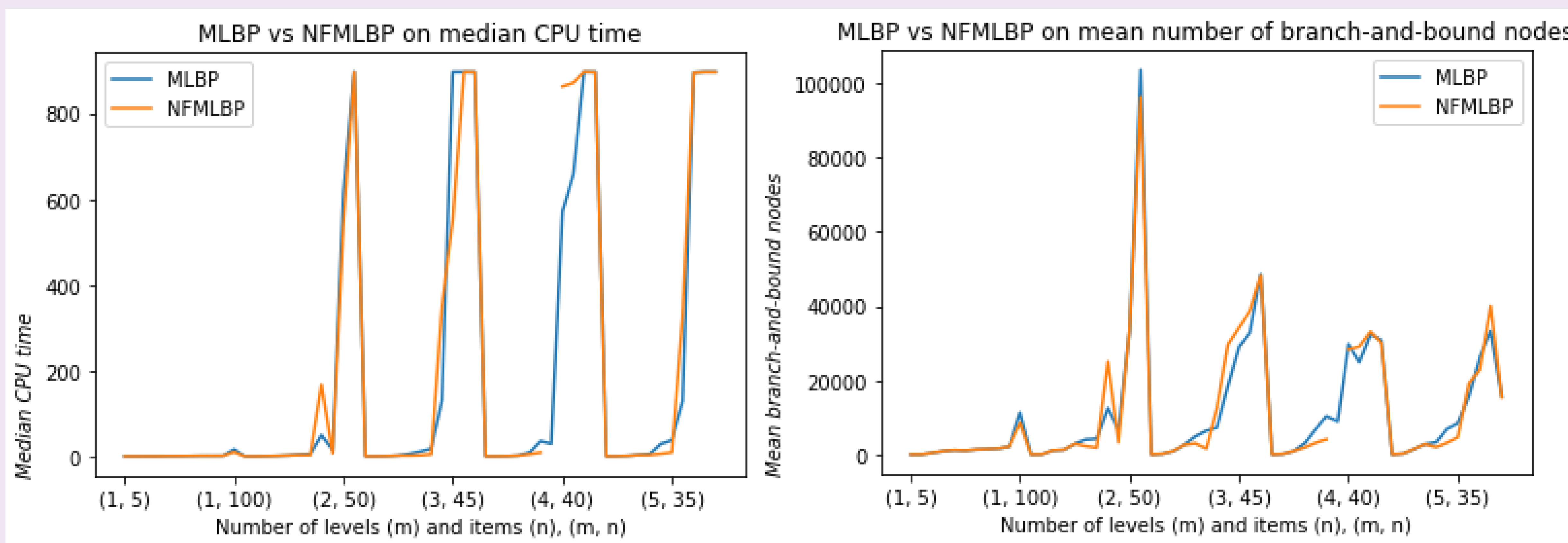

Fig. 2 Comparison of MLBP and NFMLBP, median CPU time and mean nr. of branch-and-bound nodes

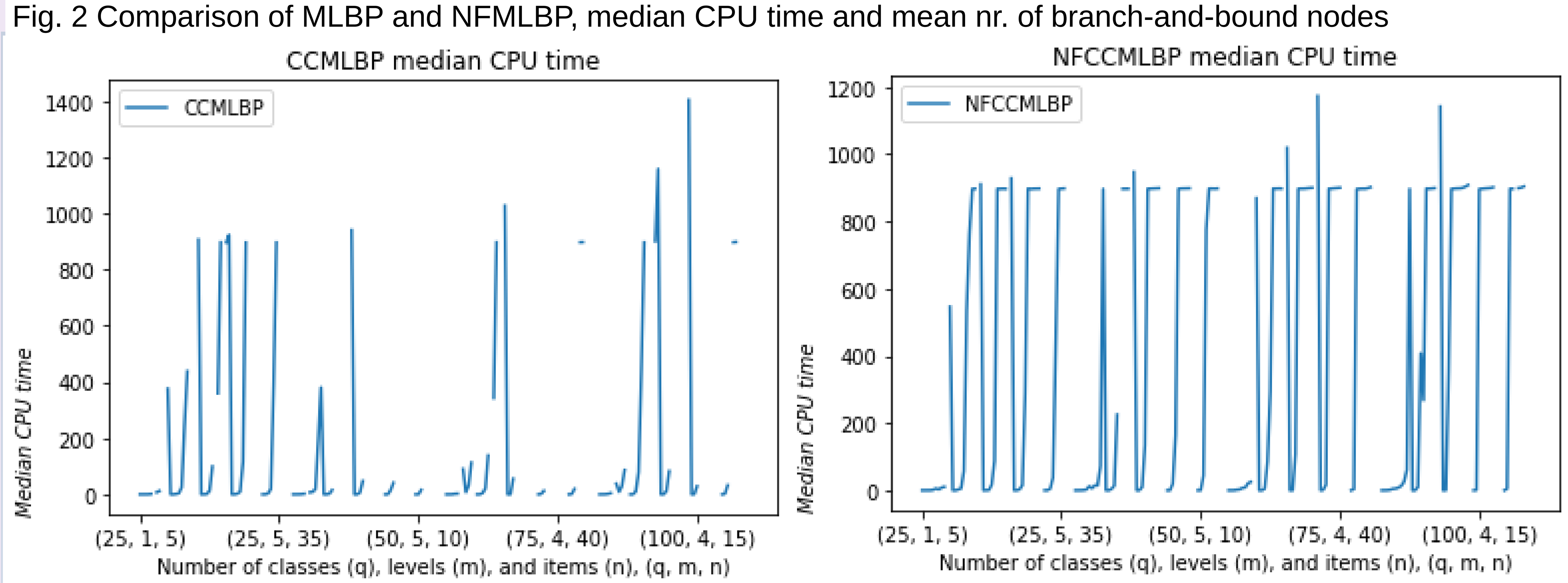

Fig. 3 Comparison of CCMLBP and NFCCMLBP, median CPU time

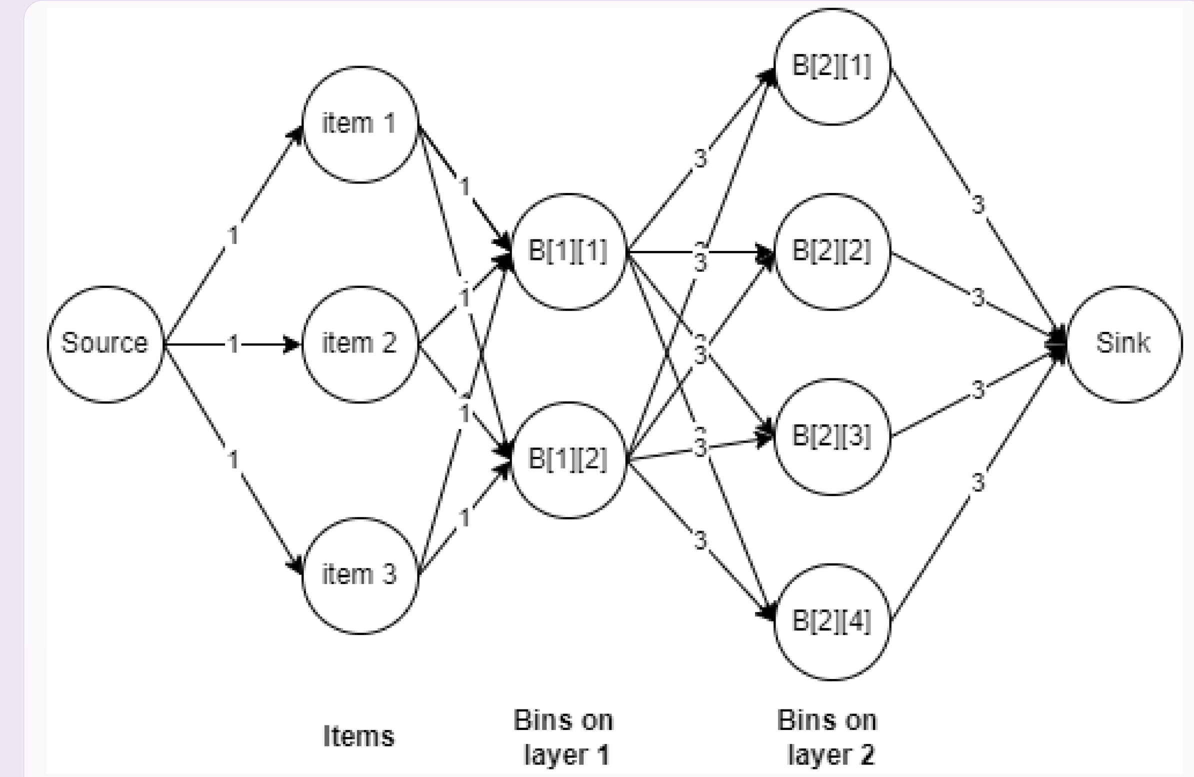

Fig. 1 A representation of a small instance of the network flow formulation. (3 items, 2 levels)

5. RESULTS

- For MLBP/NFMLBP, with the standard formulation **11% more** optimally solved instances. (77% vs. 66%).
- NFMLBP required more branch and bound nodes on average. (Fig. 2)
- MLBP/NFMLBP: up to **5 levels 35 items.** Under 5 levels, comfortably up to 40 items. (Fig. 2)
- For CCMLBP/NFCCMLBP, with the NF formulation **6% more** (36% vs. 43%)instances solved to optimality and largely more to **feasibility**. (Fig. 3)
- CCMLBP/NFCCMLBP: up to **5 levels 10 items.** Under 5 levels, comfortably up to 20 items.

6. CONCLUSTION

- Even though NF formulation performed worse in the simple MLBP problem, it still may be a promising addition due to more robustness in CCMLBP case.
- More insight on optimizations applied by CPLEX needed.
- Overall, a relatively large size of instances of MLBP/CCMLBP can be efficiently solved using the IP approach and CPLEX.