Colored Bin Packing - Checkpoint

Louis Luo

March 1, 2021

1 Introduction

This paper studies the offline versions of the Colored Bin Packing Problem (CBPP). As in the classic bin packing problem, given a set of indivisible items $N = \{1, ..., n\}$, each item is characterized by a size w_i, and the goal is to pack them into the minimum number of bins of size W. In addition, each item i is of color $c_i \in G$, where $G \subseteq \mathbb{N}$ is the set of all colors. A feasible solution for the problem consists of partitioning N into sets $N_1, N_2, ...$ such that each partition is a bin and within each partition, no two items of the same color are packed consecutively.

In this paper, exact methods for both unrestricted and restricted offline cases of CBPP are proposed. Note that a trivial upper bound on the number of bins is $|I|$, and hence the set of bins initialized for all models in this document can be denoted J, where $J = I$. This trivial upper bound will be used across all models in both cases of the problem. The item sizes are assumed to be integers.

2 Literature Review

The Colored Bin Packing was first introduced in the form of its special case called the Black and White Bin Packing Problem (BWBPP) by Balogh et al. [1, 2], where each item can only take on one of the two colors in $G = \{Black, White\}$. Since then, three settings of the Colored Bin Packing has been distinguished:

> **Unrestricted Offline**: The entire set of items is known in advance and they are given as an unordered set.
>
> **Online**: The items are presented one by one according to some list and no future information is given nor known in advance.
>
> **Restricted Offline**: The entire list of items is known in advance. Items are packed sequentially according to their respective order in the list.

Most of the work on both BWBPP and CBPP has focused on approximation algorithms. For the restricted offline and online cases of BWBPP, Balogh et al. [1, 2] proposed an algorithm with an absolute competitive ratio of 3 in the general case and proved that no online algorithm has an asymptotic competitive ratio smaller than 1.7213. For the unrestricted offline case, they show that there exists a 2.5-approximation algorithm. Furthermore, they showed that if all items have zero weights, the optimal solution is trivially found for all three cases of BWBPP. Bohm et al [3, 4] was the first to study the online version of the CBPP, and proposed an algorithm with competitive ratio of 4. They also studied the special case when all items are of size zero and found that the minimum bins used is equal to the color discrepancy, and proposed an optimal algorithm for that special case.

Many other conflict-based extensions of the bin packing problem have been studied. Bin Packing with conflicts, introduced by Jansen [5], extends standard bin packing by satisfying a conflict graph on the items with an edge denotes a pair of items that cannot be placed in the same bin. Several methods, such as branch-and-price [6] and large neighborhood search [7], have been proposed. Another variation is the co-printing problem, where in addition to the bin packing framework, each bin has a maximum number of colors across assigned items. Heuristics and a branch-and-price model [8] have been successfully employed. Bergman et al. [9] solved a related problem called Bin Packing with Minimum Color Fragmentation, where in lieu of a bin minimization objective and the color conflict, the total count of unique colors within a bins is minimized. They proposed a direct mixed-integer programming formulation and a network-flow based mixed-integer programming formulation based on binary decision diagrams.

To the best of author's knowledge, no formal attempts has been made at considering exact methods for either offline versions of GCPP.

3 Restricted Offline GCPP

Given a list of items that needs to be packed into a bin, the restricted offline GCPP requires the packing of the items to be sequential with respect to their index order.

Observation 3.1. The input order of the list is respected if and only if, within each bin, the items are packed with respect to that order.

3.1 Constraint Programming

A constraint programming model that assigns orders to bins is considered.

Variable	Description	Domain
X_i	Integer variable indicating the index of the bin that item i is assigned to	N
CP_j	Integer variable for the capacity of bin j	$0, ..., W$
CC_{jc}	Integer variable for the number of occurrences of color c in bin j	N

Table 1: Decision Variables for Constraint Programming model for the restricted offline case

$$\text{minimize } max(\mathbf{X})$$

$$\text{s.t} \quad pack(\mathbf{CP}, \mathbf{X}, \mathbf{w})$$

$$CP_j \leq CP_{j+1} \quad \forall j \in J \setminus \{|J|\}$$

$$gcc(< CC_{jc} | \forall j \in J >, < X_i | \forall i \in N s.t c_i = c >, < j | \forall j \in J >) \quad \forall c \in G$$

$$X_k = X_l \implies any(< (X_i = X_k) | \forall i \in \{\{k+1, ..., l\} | c_k \neq c_i\} >) \forall (k, l) \in \{N \times N | c_k = c_l \wedge k \leq l\}$$

The last constraint models the relationship that if two items of the same color are assigned to the same bin, then there must another item in between their positions in the input list that are assigned to that bin as well.

3.2 Mixed-Integer Programming

Variable	Description
X_{ij}	Binary variable equaling 1 iff item i is assigned to bin j
b_j	Binary variable equaling 1 iff bin j is used

Table 2: Decision Variables for Mixed-Integer Programming model for the restricted offline case

$$\text{minimize } \sum_{j \in J} b_j$$

$$\text{s.t} \quad b_j \geq b_{j+1} \quad \forall j \in J \setminus \{|J|\}$$

$$b_j \geq x_{ij} \quad \forall i \in N, j \in J$$

$$\sum_{j \in J} x_{ij} \leq 1 \quad \forall i \in N$$

$$\sum_{i \in N} x_{ij} * w_i \leq W \quad \forall j \in J$$

$$\sum_{i \in I} x_{ij} \leq x_{kj} + x_{lj} - 1 \forall I \subseteq \{x \in [k, ..., l] | c_x \neq c_i\}, (k, l) \in \{N \times N | c_k = c_l \wedge k \leq l\}$$

Again, the last constraint models color conflicts between items assigned to the same bin.

4 Unrestricted Offline GCPP

The unrestricted offline GCPP is identical to the restricted offline GCPP except that the packing of the input items does not need to adhere to any particular order, as long as the conflict constraints are satisfied.

Proposition 4.1. Given a bin and a set of items within that bin, there is a feasible packing that satisfies the color conflict if and only if the set of items can be partitioned into two subsets S1 and S2 such that $0 \leq card(S1) - card(S2) \leq 1$ and for every item in S2, there exists an item in S1 with a different color.

Proof. Idea: Every pair of items from S1 and S2 can be linked together into a sequence. Any conflict due to the colors at the linking of each pair can be resolved by switching the items in the pair. □

4.1 Constraint Programming

A constraint programming model that assigns orders to bins is considered.

Variable	Description	Domain
X_i	Integer variable indicating the index of the bin that item i is assigned to	N
CP_j	Integer variable for the capacity of bin j	$0, ..., W$
CC_{jc}	Integer variable for the number of occurrences of color c in bin j	$0, ..., N$

Table 3: Decision Variables for Constraint Programming model for the unrestricted offline case

$$\text{minimize } max(\mathbf{X})$$

$$\text{s.t} \quad pack(\mathbf{CP}, \mathbf{X}, \mathbf{w})$$

$$CP_j \leq CP_{j+1} \qquad \forall j \in J \setminus \{|J|\}$$

$$gcc(< CC_{jc} | \forall j \in J >, < X_i | \forall i \in N s.t c_i = c >, < j | \forall j \in J >) \forall c \in G$$

$$\frac{1}{2}(\sum_{c \in G} CC_{jc} + (\sum_{c \in G} CC_{jc})\%2) \leq \max_{c \in G}(CC_{jc}) \qquad \forall j \in J$$

The last constraint manifests Proposition 4.1.

4.2 Mixed-Integer Programming

Variable	Description
X_{ij}	Binary variable equaling 1 iff item i is assigned to bin j
b_j	Binary variable equaling 1 iff bin j is used
k_{qj}	Integer variable denoting the number of items in bin j with color q

Table 4: Decision Variables for Mixed-Integer Programming model for the unrestricted offline case

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j \in J} b_j \\
\text{s.t} \quad & b_j \geq b_{j+1} & \forall j \in J \setminus \{|J|\} \\
& b_j \geq x_{ij} & \forall i \in N, j \in J \\
& \sum_{j \in J} x_{ij} \leq 1 & \forall i \in N \\
& \sum_{i \in N} x_{ij} * w_i \leq W & \forall j \in J \\
& k_{qj} = \sum_{i \in \{I | c_i = q\}} X_{ij} & \forall j \in J, q \in G \\
& \sum_{q \in G} k_{qj}/2 \geq \max_{q \in G} k_{qj} & \forall j \in J
\end{aligned}
$$

Again, the last two constraints manifest Proposition 4.1 and ensure items can be packed such that the color constraint is satisfied.

5 Preliminary Results

The parameters to the problem are:

- Number of Colors: [16, 32, 64]
- Number of Items: [16, 32, 64]
- Maximum Item Size: [W, 0.75 W, 0.5 W, 0.25 W]
- Bin Capacity: [16]

Data instances are randomly generated according to the combinations of the parameters.

5.1 Restricted Offline GCPP

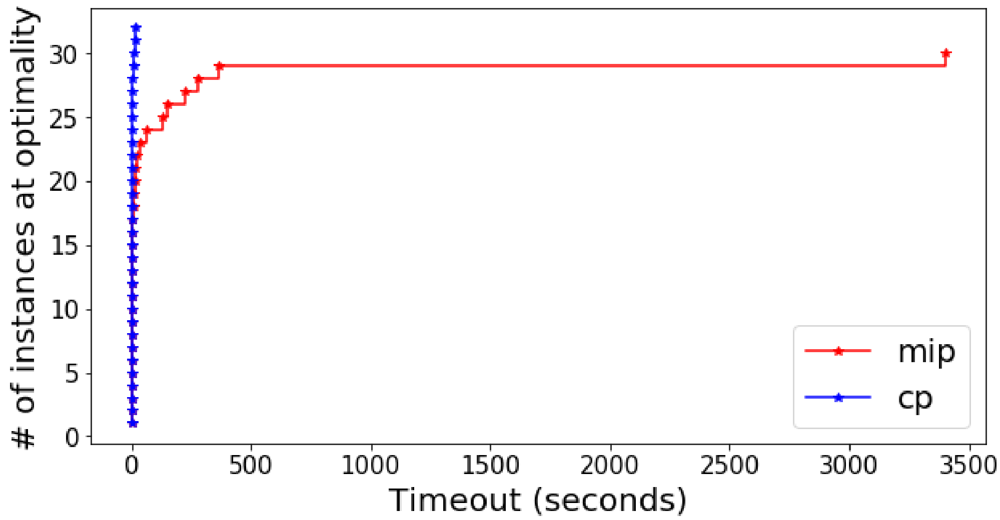

Figure 1: Caption

5.2 Unrestricted Offline GCPP

The Mixed-integer model failed to solve a lot of the instances due to memory issue. The Constraint Programming model seemed fairly robust with respect to scalability in comparison.

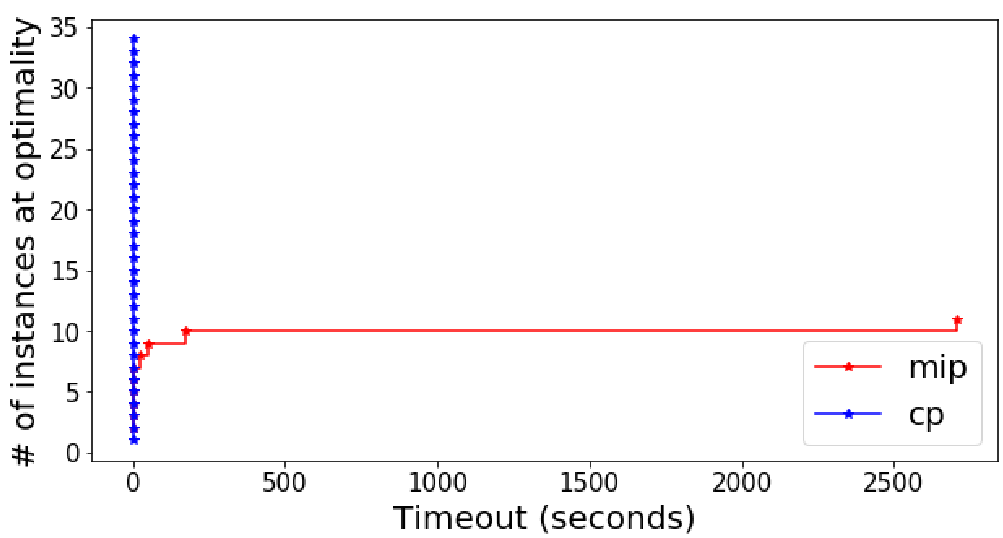

Figure 2: Caption

6 Next Steps

In the forthcoming weeks, I intend to construct the proof for the proposition in section 4. I also intend to run more experiments to understand the scalability of the current models and create additional datasets. Another interesting consideration is to reconstruct the online algorithms in other studies for the GCPP problem to compare performance. Furthermore, constructing a stronger upper bound $|J|$ on the number of bins may help MIP models that ran out of memory.

In addition, I intend to construct a SAT-based / SMT-based model from the models presented for both restricted and unrestricted versions of the GCPP problem. In addition to a direct encoding of the model formulation, I would like to formulate an order encoding formulation that considers the sequential order between items.

References

[1] J. Balogh, J. Békési, G. Dosa, H. Kellerer, and Z. Tuza, "Black and white bin packing," in *Approximation and Online Algorithms*, T. Erlebach and G. Persiano, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 131–144.

[2] J. Balogh, J. Békési, G. Dósa, L. Epstein, H. Kellerer, A. Levin, and Z. Tuza, "Offline black and white bin packing," *Theoretical Computer Science*, vol. 596, pp. 92–101, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304397515005551

[3] M. Böhm, J. Sgall, and P. Veselý, "Online colored bin packing," in *Approximation and Online Algorithms*, E. Bampis and O. Svensson, Eds. Cham: Springer International Publishing, 2015, pp. 35–46.

[4] M. Böhm, G. Dósa, L. Epstein, J. Sgall, and P. Veselý, "Colored bin packing: Online algorithms and lower bounds," 2018.

[5] K. Jansen, "An approximation scheme for bin packing with conflicts," *Journal of Combinatorial Optimization*, 1999.

[6] R. Sadykov and F. Vanderbeck, "Bin packing with conflicts: A generic branch-and-price algorithm," *Informs*, 2012.

[7] R. Capua, Y. Frota, L. S. Ochi, and T. Vidal, "A study on exponential-size neighborhoods for the bin packing problem with conflicts," *Journal of Heuristics*, vol. 24, 2019.

[8] A. Kondakov and Y. Kochetov, "A core heuristic and the branch-and-price method for a bin packing problem with a color constraint," *Springer International Publishing*, 2018.

[9] D. Bergman, C. Cardonha, and S. Mehrani, "Binary decision diagrams for bin packing with minimum color fragmentation," *CPAIOR*, 2018.