A BALANCED BRANCHING STRATEGY FOR SET PACKING PROBLEMS

Barış Evrim Demiröz
San Jose, CA
baris.evrim.demiroz@gmail.com

May 11, 2020

ABSTRACT

Solving linear programming relaxations of weighted set packing problems might yield fractional solutions. A common approach to find the optimal integer solution is using a search tree with the branch-and-bound algorithm. The performance of the search depends on the balancedness of the search tree. In this work I present a branching strategy for weighted set packing problems that leads to a more balanced search tree.

***K*eywords** set packing · branch-and-bound · combinatorial optimization

1 Introduction

Set packing is a classical problem of combinatorial optimization along with set covering and partitioning. Weighted variant of set packing problem can be defined as

$$\begin{array}{ll} \text{max} & \mathbf{w}^{\mathsf{T}}\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \leq \mathbf{1} \\ & \quad \mathbf{x} \in \{0,1\}^n. \end{array}$$

A straightforward way of solving this problem is relaxing the integrality constraint and solving it as a linear program (LP). However, linear relaxation of the same problem might not yield an integer solution. In that case, an integer solution is usually sought by splitting the solution space into preferably disjoint sets at each iteration and pruning paths early using an upper bound. This approach is called *branch-and-bound* [1].

A simple way of branching is creating two subproblems by rounding a fractional valued variable up and down. However, this leads to an unbalanced search tree [2]. Say, the fractional variable is x_i. In one branch, selected subset ($x_i = 1$) possibly prevents other subsets from being selected thus restricting the solution space, while in the other branch only one subset is excluded ($x_i = 0$) leaving many candidate solutions to be explored.

Creating a balanced search tree for branch-and-bound algorithm is important for reaching an optimal solution quickly. Ryan and Foster proposed a branching strategy that leads to balanced search trees for set partitioning problems [3]. It is possible to convert set packing problem to a partitioning problem by adding slack variables and converting inequality constraints to equality. This enables using Ryan and Foster branching strategy for set packing problems. Yet, in practice this might require adding too many variables and the resulting problem is computationally too expensive to solve.

In this work I propose a Ryan and Foster like branching strategy for set packing problems. This scheme was first reported in [4] where the subsets to be packed are rectangles on a 2D domain. However in that work, the results obtained using a different branching strategy were inconclusive. In this paper I present a new generalized form of the branching strategy and empirically show that it leads to significantly fewer branchings for weighted set packing problems compared to the conventional strategy of rounding variables. I also perform a statistical analysis on the results reported in [4] and quantify the significance.

2 Branching strategy

First, let's start with an observation. Our branching strategy will depend on this property.

Proposition 1. *Let* $\mathbf{x}$ *be the fractional optimal solution to the weighted set packing problem, i.e.* $0 < x_i < 1$ *for some* $i \in \{1, \ldots, n\}$. *There exist two rows* e, f *and a column* $j \neq i$ *of the constraint matrix* $\mathbf{A}$ *such that:*

$$\begin{aligned} a_{ei} &\neq a_{ej} \\ a_{fi} &= a_{fj} = 1 \\ x_j &> 0 \end{aligned}$$

Proof. Assume $a_{fj} = 0$ for all $f : a_{fi} = 1$ and $j : x_j > 0$ and $j \neq i$. Then we can improve the objective value by setting $x_i = 1$ since doing so does not violate any constraints. This contradicts that $\mathbf{x}$ is an optimal solution. So there must exist a column j and row f with $a_{fi} = a_{fj} = 1$. Because there are no duplicate columns in the basis, for any two columns we can find a row e such that $a_{ei} \neq a_{ej}$. □

After identifying rows e and f we can impose the branching constraints

$$\sum_{i:a_{ei}=a_{fi}=1} x_i \leq 0 \quad \text{and} \quad \sum_{i:a_{ei}=a_{fi}=1} x_i \geq 1$$

for left and right branches respectively. This means, the elements e and f have to be covered by different subsets on the left branch and by the same subset on the right branch.

As a consequence of the proposition above, if it is not possible to identify any (e, f) pairs, then the solution of the weighted set packing problem must be integer. The algorithm terminates after a finite number of branching since there are only a finite number of row pairs.

3 Experiments

In this section two branching strategies are compared. The first one is the conventional branching strategy where a fractional variable is rounded up and down at each branch. The other one is the novel branching strategy proposed in this work.

3.1 Synthetic problems

Two branching strategies are compared on artificially generated weighted set packing problems. In total, 100 weighted set packing problems are generated. For each problem, number of variables, n, are uniformly generated to be between 50 and 10; number of constraints are uniformly generated to be between n and $n + 100$. Weights are uniformly generated to be between -1 and 0.

Each entry in the constraint matrix is generated using Bernouilli distribution with 0.5 as the parameter. As a result, almost all of the problem instances resulted in fractional solutions for their LP relaxations. If LP relaxation of the problem yields integer solutions, the problem is discarded and a new problem is generated.

For all generated weighted set packing problems, proposed branching scheme resulted in fewer number of visited nodes. The amount of reduction in the number of visited nodes vary between 1.9 and 5.2 fold.

The code to reproduce the experiments are made publicly available[1].

3.2 Rectangle Blanket Problem

Here, the results of the Rectangle Blanket Problem (RBP) experiments that was conducted in [4] are reused. RBP is a weighted set packing problem in its core. The authors solve it by using *branch-and-price* algorithm, which is a branch-and-bound algorithm in a column generation setting. In Table 1 two branching strategies are compared.

I have performed a Wilcoxon signed-rank test to see if the novel method is better in terms of number of nodes visited in the search tree. Wilcoxon signed-rank test is a non-parametric statistical hypothesis test. Although the result is not very significant (p value of $.117$) it is promising. The novel method does not always win over the conventional method probably because the strategy is used in a branch-and-price column generation scheme and there are multiple optimal solutions.

[1] It can be accessed via `https://github.com/barisdemiroz/set_packing_balanced_branching`

Table 1: The number of visited nodes in the search tree for each problem instance. SCHEME ROUNDING is the strategy of creating branches by rounding a fractional variable up and down. SCHEME NOVEL is the proposed strategy.

instance	SCHEME ROUNDING	SCHEME NOVEL	instance	SCHEME ROUNDING	SCHEME NOVEL
dog-12-K3	73	115	toy11-K10	3	5
dog-13-K3	79	55	typical4-K10	6	6
key-16-K3	3	9	dog-6-K15	5	7
key-18-K3	3	3	toy1-K15	3	23
toy13-K3	3	351	toy10-K15	33	13
toy4-K3	89	43	toy11-K15	7	13
avatar4-K5	7	9	toy14-K15	9	3
device5-13-K5	27	27	avatar4-K20	9	3
device5-8-K5	18	41	dog-12-K20	17	17
key-17-K5	51	47	dog-13-K20	15	9
toy12-K5	5	5	dog-16-K20	7	3
toy14-K5	25	29	dog-17-K20	61	3
bat-2-K10	48	37	dog-7-K20	3	3
bat-6-K10	179	61	toy2-K20	958	5
device5-18-K10	7	3	toy4-K20	15	11
dog-12-K10	37	5	toy9-K20	3	3
dog-13-K10	7	17			

4 Conclusion

In this work I have proposed a novel branching strategy for solving weighted set packing problems using branch-and-bound. This new strategy shares similarities with Ryan and Foster's branching strategy for set partitioning. Initial experimental results show this is a promising approach that is worth further investigation possibly on other set packing problems.

References

[1] Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

[2] François Vanderbeck. Implementing mixed integer column generation. In *Column generation*, pages 331–358. Springer, 2005.

[3] D. M. Ryan and B. A. Foster. An integer programming approach to scheduling. In A. Wren, editor, *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*, pages 269–280. North-Holland, Amsterdam, 1981.

[4] Barış Evrim Demiröz, İ Kuban Altınel, and Lale Akarun. Rectangle blanket problem: Binary integer linear programming formulation and solution algorithms. *European Journal of Operational Research*, 277(1):62–83, 2019.