

A steepest edge rule for a column generation approach to the convex recoloring problem

Ergin Erdem¹, Kenneth Gahler^{1***}, Eunseok Kim^{2†} and Sangho Shim¹

¹ Department of Engineering, Robert Morris University, Moon Twp, PA 15090, USA,
shim@rmu.edu,

² KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon, 34141, Republic of Korea

Abstract. The convex recoloring problem is a clustering problem to partition nodes of a network into connected subnetworks. We develop a hybrid rule combining the Dantzig’s Rule and the Steepest Edge Rule to produce columns which enter into the basis of the master problem in the column generation framework introduced by Chopra et al. [2]. The hybrid rule leads to only a small number of iterations and makes it possible to perform the column generation approach in an undergraduate class using Microsoft Excel. We perform a large scale computational experiment and show that the hybrid rule is effective.

1 Introduction

A column generation approach performs the simplex method to solve a huge scale of linear programming problem which we call the *master problem*. While a general linear programming approach enumerates the reduced costs of the columns which measure the contribution of the columns toward the optimal solution, the column generation approach keeps and updates only a small set of columns, which we call a *basis*, without enumerating the columns. Instead, the *subproblem* of the column generation approach generates a column which improves the current basis most effectively by replacing the least effective column of the basis.

Johnson, Mehrotra and Nemhauser [7] first developed a column generation framework for a clustering problem, and Chopra, Erdem, Kim and Shim [2] adjusted the framework to the convex recoloring problem to recolor nodes of a colored graph at minimum number of color changes such that each color induces a connected subgraph where a pair of nodes colored with a same color are connected by a path of nodes colored with the same color. Equivalently, the convex recoloring problem is to maximize the number of nodes where the initial colors stay without change.

*** Kenneth Gahler is an undergraduate student of Robert Morris University. He performed the small scale of computational experiment using Microsoft Excel in Section 3.1

† Eunseok (Isaac) Kim is a Robert Morris University visiting graduate student from KAIST (Korea Advanced Institute of Technology and Science, South Korea). Isaac

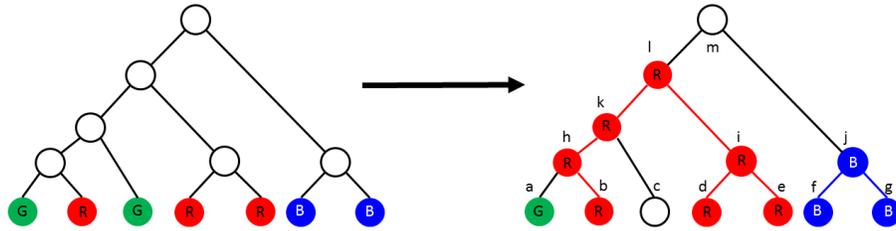


Fig. 1. Convex recoloring

Max	3	1	2	0	0	...	= 6
R	1					...	= 1
G		1				...	= 1
B			1			...	= 1
a		1				...	= 1
b	1					...	= 1
c				1		...	= 1
d	1					...	= 1
e	1					...	= 1
f			1			...	= 1
g			1			...	= 1
h	1					...	= 1
i	1					...	= 1
j			1			...	= 1
k	1					...	= 1
l	1					...	= 1
m					1	...	= 1

Fig. 2. The master problem

Figures 1 and 2 illustrate an optimal solution to the convex recoloring problem on a phylogenetic tree and the columns in an optimal basis of the master problem. In Figure 1, given the coloring at the leaf nodes on the left, each color makes a connected subgraph in the optimal solution on the right which is obtained by changing only one color at node c or keeping the maximum number of initial colors at the other six leaf nodes. Note that we can color uncolored nodes at no cost but uncolor an initial color at the unit cost. Figure 2 depicts the columns of an optimal basis in the coefficient matrix A of the master problem

$$\max\{WZ : AZ = \mathbf{1}, Z \geq 0\},$$

where $\mathbf{1}$ denotes the vector of all components 1. The first components of a column A_j indicate a color and the next components indicate the nodes colored with the color. The objective coefficient W_j corresponding to A_j is the number of nodes where the initial color remains the same one as indicated by the first components. The binary variable $Z_j = 1$ will indicate that A_j is in the basis. The basis is updated at every iteration by a best contributing column which is generated by a small size of integer programming.

In the column generation framework introduced by Chopra et al. [2], the subproblem to generate the best contributing column is solved in two steps:

- Step 1. For each color, generate the column of the maximum reduced cost which indicates nodes colored with the color.
- Step 2. Enumerate the generated columns for the colors and pick the best one.

Dantzig's rule selects the column of the best reduced cost and is used in Step 1. Chopra et al. [2] used Dantzig's rule in Step 2 as well. Since the columns generated in Step 1 can be enumerated in Step 2, we can apply the steepest edge rule which is suggested by Goldfarb and Reid [6]. The steepest edge rule picks the column which leads the basis in the direction of the sharpest angle with the gradient W of the objective function. This partial use of the steepest edge rule reduces the number of iterations and allows us to implement the column generation approach using Microsoft Excel.

In Section 2, we discuss the pedagogy of column generation approaches and the steepest edge rule. In Section 3, we perform computational experiments comparing Dantzig's rule with the hybrid rule. We conclude that the hybrid rule works well across small and large numbers of colors while Dantzig's rule only does not perform well in a small number of colors.

2 Pedagogy

2.1 Column generation

The general linear programming problem was formulated in the late 1940s, and the simplex algorithm for solving it was proposed by George Dantzig in 1947.

performed the large scale computational experiment running his JAVA code in Section 3.2

The field of linear programming developed simultaneously with computing technology. In recent years both areas have experienced very rapid development. Computers have become much more powerful, and much less expensive. And linear programming software has become much more widely available; for example, Excel Solver is an add-in program which is a built-in program of Microsoft Excel.

Linear programming is taught as a core course in every academic program of Industrial Engineering. Since linear programming software are available everywhere, students raise this question; “Why do we have to study the simplex algorithm? We only need to use the software.”

The instructor may answer the question with the column generation approach. No linear programming software immediately performs the column generation approach, while the column generation approach is used in almost every industry; in particular, it is used for a large scale of scheduling in transportation and logistics such as airline industry and rail freight industry. It can tackle a variety of large scale optimization problems using only a small size of memory.

Column generation approach was first developed by Gilmore and Gomory [4, 5] to solve the cutting stock problem. The cutting stock problem is to minimize the number of raws that are cut into finals to satisfy the customer orders. Since many sizes of finals are required to satisfy customer demands, a company would like to satisfy demands by cutting up the least possible number of raws. The cutting stock problem arises in industries that produce materials such as paper, textiles, and sheet-metal. Such products are often manufactured in large rolls called raws. These rolls are then cut into smaller rolls called finals to satisfy customer orders.

The cutting stock problem can be modeled as a linear programming formulation, called the master problem, including a huge population of columns which indicate all the cutting patterns. The beauty of the linear programming model is that we only need to keep and update a small number of columns which we call as basis. In practice a column generation procedure can deal with the master problem by producing a relatively short list of patterns, each of which appears likely to contribute to the optimal solution of the master problem. The column generation procedure itself presents a minor difficulty in that it requires the solution of a small size of integer programming problem which we call the subproblem. The solution to the subproblem is the column which enters into the basis replacing a column.

A column generation approach updates a small set B of columns which we call as basis. In each iteration, the sub-problem identifies a column of the optimal reduced cost from the huge pool A of the columns of the master problem. The reduced cost of a column is a measure of the column’s contribution to improving the objective value of the master problem. The rule of updating the basis with the column of the best reduced cost is referred to as Dantzig’s Rule. Since the reduced cost is a linear objective function in a column generation framework, the subproblem is modeled as an integer programming formulation; *i.e.*, a linear programming formulation with integer variables. The integer programming

prices out the reduced costs of the columns of the master problem without enumerating them and generates the entering column of the best reduced cost as the optimal solution.

2.2 Steepest Edge Rule

Suppose we are solving a linear programming problem $\max\{wx : Ax = b, x \geq 0\}$ by the simplex algorithm, and that we have arrived at a vertex v_0 on the constraint polyhedron. In general it will be possible to improve the objective function by moving along any one of several edges leading out of v_0 , to a neighboring vertex v_i . Figure 3 depicts this situation. In this case the objective function can be improved by moving to either v_1 , v_2 or v_3 . Among the neighboring vertices, v_1 is of the sharpest angle with w ; *i.e.*, the angle between $v_1 - v_0$ and w is minimum. The steepest edge rule chooses the entering column to be the one that results in a move to the vertex v_i with the sharpest angle with w .

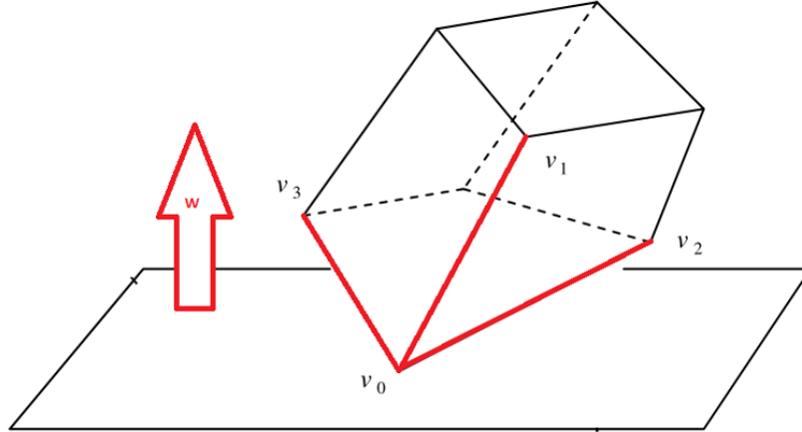


Fig. 3. Geometry of the steepest edge rule

It is a consensus among practitioners that the steepest edge rule performs significantly better than Dantzig's rule against degeneracy and cycling; *i.e.*, it rarely occurs with the steepest edge rule for a same sequence of columns to enter repeatedly into the basis without improving the objective value. Unfortunately, the steepest edge rule has not been used for a column generation framework, as the measure of the contribution of an entering column is not linear in the steepest edge rule. The column generation framework for the convex recoloring problem allows us to use the steepest edge rule along with Dantzig's rule. For each color,

the subproblem is to generate the best column in the reduced cost by Dantzig's rule. Among the generated columns, we enumerate and pick the best one in the steepest edge rule. This partial use of the steepest edge rule reduces the number of iterations and allows us to implement the column generation approach in an undergraduate class using Microsoft Excel.

3 Computation

We perform computational experiments comparing Dantzig's rule with the hybrid rule. We perform the experiment over small scale of problem instances using Microsoft Excel and over large scale ones running JAVA code where the subproblem is solved by a fast dynamic programming algorithm introduced by Chopra et al. [3].

3.1 Small Scale Clustering using Microsoft Excel

A tree is a connected acyclic graph and a connected subgraph of a tree is a subtree. For each color t , the subtree problem is to identify the subtree colored with t of the maximum reduced cost, and is formulated as

$$\max_x \left\{ \sum_{v \in V} (w(v, t) - \pi_v) x_v : x \text{ is a subtree} \right\},$$

where binary variables $x_v \in \{0, 1\}$ indicate the nodes of a subtree. We employ additional binary variables $y_e \in \{0, 1\}$ which indicate both end nodes of edge e are colored with a same color. We can formulate the subproblem as follows:

$$\max \sum_{v \in V} (w(v, t) - \pi_v) x_v \quad (1)$$

$$s.t. \sum_{u \in V} x_u - \sum_{e \in E} y_e \leq 1 \quad (2)$$

$$\left. \begin{array}{l} -x_u + y_{uv} \leq 0 \\ -x_v + y_{uv} \leq 0 \end{array} \right\} \text{ for edge } uv \in E \quad (3)$$

$$x_u \in \{0, 1\} \text{ for } u \in V \quad (4)$$

$$y_e \in \{0, 1\} \text{ for } e \in E \quad (5)$$

As a tree is a generalization of a path, (2) says that the number of nodes is one more than the number of edges. We see that (2) is a necessary condition for a subtree. It is also a sufficient condition because the subgraph is on a tree which has no cycle. Inducing a subgraph is described by (3).

The problem instance in Figure 1 is solved using Excel Solver in 5 iterations with the hybrid rule and in 6 iterations with Dantzig's rule. The number of iterations with the hybrid rule is observed to be uniformly smaller than that with Dantzig's rule. With Dantzig's rule only, cycling occurred frequently and we could not achieve the optimal solution.

3.2 Advanced Computing

Chopra et al. [2] developed a column generation approach to the convex recoloring problem on a tree and successfully solved those problem instances which the integer programming introduced by Chopra et al. [3] could not solve. If the number of colors is large, the column generation approach performed extremely fast. However, if the number of colors is small, the number of elements is large on average and the solutions are also highly degenerate.

Node#	Color#	CG_TotalElabTime	Rev_Simplex_TotalElabTime
636	6	1136.098265	20.02831585
636	35	2.485365974	31.28726364
636	333	12.97899625	612.0650301
636	7	7201.538144	29.34062022
636	28	7203.495308	37.20071777
636	317	14.18405912	597.0863872
636	5	7202.972239	14.45203846
636	28	2.381203368	38.71373915
636	309	16.72643312	887.3193219
710	10	7205.444582	18.91777751
710	35	7202.983335	31.0576981
710	361	18.0937688	873.1824033
710	4	7203.809618	9.947785205
710	41	7204.897622	35.72443954
710	338	18.51892624	605.7163418
710	6	7201.991398	24.37458391
710	36	63.05581796	45.05815185
710	361	19.38627406	750.6392457

Fig. 4. Computational time (sec.) of Dantzig’s rule vs. our hybrid rule

With the hybrid of Dantzig’s rule and the steepest edge rule, we can resolve this problem. The table in Figure 4 compares the hybrid rule with Dantzig’s rule over phylogenetic trees from TreeBASE.org. The first column is the number of nodes of the phylogenetic tree and the second column is the number of colors. In the third column we see the elapse time of Dantzig’s rule frequently exceeds 2 hours (7,200 seconds) in case of a small number of colors. The fourth column shows that our hybrid rule is strong against the small number of colors and the dense basis.

4 Conclusion

We developed a hybrid rule so that the steepest edge rule can be partially used to overcome degeneracy and cycling which had been a systematic problem in a column generation framework for the convex recoloring problem. In computational experiments over small scale experiments and over large scale experiments, we observed that the hybrid rule performs well across small and large numbers of colors.

Acknowledgement. For summary of column generation approach and the steepest edge rule, we refer to Lecture Notes on Computational Methods by Earl Barnes [1] at Georgia Institute of Technology.

References

1. E. Barnes, Lecture Notes on Computational Methods, Georgia Institute of Technology.
2. S. Chopra, E. Erdem, E. Kim and S. Shim, Column generation approach to the convex recoloring problem on a tree, *Modeling and Optimization: Theory and Applications (MOPTA, Bethlehem, PA, USA, August 2016)*, Volume 213 of the series Springer Proceedings in Mathematics & Statistics, pp 39-53, 2017.
3. S. Chopra, B. Filipecki, K. Lee, M. Ryu, S. Shim and M. Van Vyve, The convex recoloring problem on a tree, *Mathematical Programming* **165** (2017) 529-548.
4. P.C. Gilmore and R.E. Gomory, A Linear Programming Approach to the Cutting-Stock Problem, *Operations Research* **9** (1961) 849-859.
5. P.C. Gilmore and R.E. Gomory, A Linear Programming Approach to the Cutting Stock Problem-Part II, *Operations Research* **11** (1963) 863-888.
6. D. Goldfarb and J.K. Reid, A practical steepest-edge simplex algorithm, *Mathematical Programming* **12** (1977) 361-371.
7. E.L. Johnson, A. Mehrotra and G.L. Nemhauser, Min-cut clustering, *Mathematical Programming* **62** (1993) 133-151.