



INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Note from the Editor

To cite this article:

(2024) Note from the Editor. INFORMS Journal on Computing 36(6):1359-1361. <https://doi.org/10.1287/ijoc.2024.ed.v36.n6>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2024, INFORMS

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Note from the Editor

<https://doi.org/10.1287/ijoc.2024.ed.v36.n6>

Copyright: © 2024 INFORMS

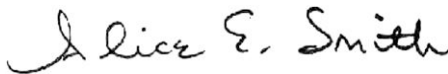
In October at the INFORMS Annual Conference in Seattle, we held our annual *IJOC* editors' meeting. The link below provides access to the report prepared by me for this event along with the accompanying PowerPoint set. The PowerPoint set was also shown at the annual INFORMS Computing Society business meeting at the same conference. If you look over these documents, you will see that our journal has an exceedingly good status. Submissions are up, downloads are up, selectivity is up, and review times are down. We have one special issue published and have two upcoming in 2025, and our *IJOC* GitHub repository is now implemented across the entire journal to build a permanent record of data, code, and other digital artifacts of the published *IJOC* papers: all fully open access and licensed.

Link to *IJOC* Annual Report: https://www.dropbox.com/scl/fo/9rrdzxiy1st5r96k9gf4/ABucEk_c8-LlrDZKED9yanU?rlkey=grho7rl3kuffa5ll1ciz869bf&dl=0.

We continue to announce the winners of *INFORMS Journal on Computing (IJOC)* Test of Time Paper Award to cover the backlog of awards since the journal's inception. The energetic and able committee chaired by John Chinneck with members Bill Cook, Bruce Golden, Pascal Van Hentenryck, and David Woodruff has selected the awardee, covering the period 1995 through 1999. What follows is the citation from the award committee and then a reflective interview with the authors concerning this paper.

I want to thank the committee for their superb efforts and am very pleased to share this recognition of the impactful heritage of our journal.

All my best,



The Test of Time Award for papers published in the *INFORMS Journal on Computing* in the years 1995–1999 is awarded to:

Anuj Mehrotra and Michael Trick

“A Column Generation Approach for Graph Coloring”

INFORMS Journal on Computing, Volume 8, Issue 4, Fall 1996, pp. 344–354

<https://pubsonline.informs.org/doi/10.1287/ijoc.8.4.344>

Test of Time Award Citation 1995–1999

Graph coloring has a simple polynomial-sized integer-programming formulation, but, even for modest-sized graphs, the resulting IP models are notoriously difficult to solve. This difficulty is due in large part to the symmetry in the model, where color classes are not distinguished in the IP variables. The solution proposed in the Mehrotra-Trick paper is to break the symmetry, creating a model with a variable for each potential color class. The resulting exponential-sized formulation is handled by delayed column generation combined with an ingenious branching rule, designed to maintain the original graph-coloring structure in the newly created subproblems.

The paper is beautifully written, developing and explaining difficult concepts in branch-cut-and-price methodology in an easy-to-grasp setting. Their elegant work has long served as an important template in the application of linear programming techniques to the solution of difficult optimization models. It will continue to do so for years to come.

Reflections on “A Column Generation Approach for Graph Coloring”

By Anuj Mehrotra and Michael Trick

November 2024

The paper “A Column Generation Approach for Graph Coloring” (Mehrotra and Trick 1996) had its genesis in the 1992–1993 academic year. During that year, Trick’s school, then the Graduate School of Industrial Administration (GSIA) of Carnegie Mellon University, generously let him spend most of his time at DIMACS (Center for Discrete Mathematics and Theoretical Computer Science) on Rutgers’ campus. During that year, Trick, along with David Johnson, ran a computational challenge on “Cliques, Coloring, and Satisfiability.” That effort involved collecting instances, solution codes, and other aspects in those areas and culminated in a collection of papers and, most importantly, a warehouse of instances to serve as benchmarks for solution techniques for finding cliques and colorings in graphs (Johnson and Trick 1996).

Meanwhile, Mehrotra was completing his dissertation on branch-and-price approaches for clustering problems from the Industrial and Systems Engineering (ISyE) program at Georgia Tech (Trick had graduated from that program four years earlier) and took a visiting position at GSIA, primarily replacing Trick for the year.

Branch-and-price is based on one of the fundamental, and early, methods in linear programming: column generation. As exemplified best by Gilmore and Gomory (1961), column generation is an approach to solve linear programming problems with a huge (exponential) number of columns. The Gilmore and Gomory paper concentrated on an approach for the cutting stock problem, where a number of stock items are divided into smaller pieces. This applies, for instance, when long pieces of steel need to be divided into smaller pieces for sale. In column generation, a linear program is formed where there is a variable for each pattern of cutting a stock piece into smaller pieces. This leads to a huge number of variables (one for every cutting pattern), so explicitly writing down the full linear program is not practical for even small instances of the problem. Instead, a smaller number of cutting patterns is used initially, and the variables corresponding to new patterns are then added if they could improve the solution. The adding of patterns is guided by dual prices of the main linear program. Interestingly, the subproblem to determine potential of the additional variable(s) need not be a linear program: any solvable subproblem would suffice. In the Gilmore and Gomory paper, the subproblem was a binary knapsack problem.

Branch-and-price addresses the problem of what to do if the main problem is an integer program. Such a problem does not have dual variables to guide the subproblem. But the dominant approach to integer programming, branch-and-bound, solves a series of linear programming problems: could the duals from those be used to guide a column generation approach?

During 1992–1993, and particularly the summer of 1993, we met regularly, often over bagels and coffee for breakfast, and we realized that graph coloring made an excellent area in which to explore branch-and-price. In particular, the DIMACS Computational Challenge showed that finding cliques, or equivalently independent sets, in a graph is much easier than coloring a graph. In a column generation approach to graph coloring, there is a variable for every independent set (corresponding to a set of nodes with the same color), and the main problem is choosing the minimum number of independent sets so that every vertex belongs to one of the chosen sets. The subproblem is to take duals from the main problem to find improving independent sets. This leads to a weighted independent set problem.

This approach, however, creates fractional independent sets. Branch-and-price is necessary to get a true graph coloring. First, a branching rule must be devised to remove fractional solutions from the main problem. Second, a solution method for the subproblem must be created that is consistent with that branching rule. Many obvious things do not work. For instance, the simplest branching rule in the main problem is to take a fractional independent set and branch into two new problems: either use that independent set or do not use the independent set. The first of these is easy: simply remove those nodes from the graph and solve on the resulting much smaller graph. The second, however, is problematic. Somehow the subproblem has to know not to recreate the forbidden independent set. Hence, to determine such an independent set, either a constraint has to be added to the subproblem, ruining its “weighted independent set” structure, or the subproblem has to find a potentially improving weighted independent set that does not correspond to the optimal solution for the subproblem that would reproduce the forbidden independent set. This is a challenging problem without a simple solution.

Instead, we realized that Ryan-Foster (Ryan and Foster 1981) branching would work well here. In this branching, two nodes that are covered fractionally are chosen, and two new problems are created: one where they have the same color and one where they have different colors. This creates two subproblems that do not destroy the weighted independent set structure. In the first, the two nodes are combined into one; in the second, an edge is added to the graph, precluding the nodes from having the same color. Branch-and-bound can now proceed with every subproblem of the corresponding column generation having the same weighted independent set structure.

With this basic structure, we had the hard work of making it work. Trick's experience with the DIMACS Challenge was with finding independent sets and not weighted independent sets. Methods for the weighted version had to be found. It was clear that a straightforward integer program was not going to suffice because such approaches had difficulty even with the unweighted version. Some combinatorial approach was needed and developed, drawing on previous work in Boolean optimization and other areas. For the main problem, branch-and-price was still in its infancy, so decisions had to be made on details of the branching rule (which pair of nodes to choose), initial solutions, node selection, and so on. When it all came together, the resulting approach to graph coloring was at least competitive with other approaches and often was the fast approach of the time.

We appreciate the selection of this paper for a "Test of Time" award. To this day, we see this paper as an excellent exemplar of what it takes to make the now-standard approach of branch-and-price to work: formulate the problem so that the subproblem is reasonably easy to solve and find a branching rule that does not destroy that structure. That worked well for us in graph coloring and well for the field in countless problems since.

References

- Gilmore PC, Gomory RE (1961) A linear programming approach to the cutting-stock problem. *Oper. Res.* 9(6):849–859.
- Johnson DS, Trick MA, eds. (1996) *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, October 11–13, 1993*, vol. 26 (American Mathematical Society).
- Mehrotra A, Trick MA (1996) A column generation approach for graph coloring. *INFORMS J. Comput.* 8(4):344–354.
- Ryan DM, Foster BA (1981) An integer programming approach to scheduling. *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling* (The National Academies of Sciences, Engineering, and Medicine, Washington, DC), 269–280.