Ann Oper Res (2007) 149:63–66
DOI 10.1007/s10479-006-0100-1

Revival of the Gomory cuts in the 1990's

Gérard Cornuéjols

Published online: 2 December 2006

In the early 90's, the research community was unanimous: In order to solve integer programs of meaningful sizes, one had to exploit the structure of the underlying combinatorial problem; Gomory cuts (Gomory, 1960, 1963) made elegant theory (because they did not require knowledge of the underlying structure) but were utterly useless in practice (because they did not use the underlying structure!). I will come back shortly to these widely held beliefs. A practitioner confronted with a general mixed integer linear program (MILP) who did not have the time or skill to "exploit the underlying structure" had to resort to commercial codes that had been perfected in the early 70's and had not been improved significantly in the following two decades. These branch-and-bound codes could solve small to medium size instances of MILP's, but many applications were beyond their ability. Because MILP is NP-hard, it was widely accepted that nothing much could be done about this state of affairs. The sentiment was that there was little research left to do on branch-and-bound algorithms for general MILP's.

On the other hand, branch-and-cut algorithms for structured problems generated a tremendous amount of excitement. Padberg and Rinaldi obtained spectacular results for the traveling salesman problem. The same general philosophy was applied with a varying degree of success to numerous other classes of problems. Typically, for each class, a paper or series of papers would first identify facets of the integer polyhedron (i.e. the convex hull of the feasible solutions to the integer program), then present separation algorithms or heuristics for these facets and finally report some computational results. The success of such branch-and-cut algorithms was attributed to the use of facets of the integer polyhedron. By contrast, prominent researchers had a low opinion of the practical usefulness of Gomory cuts, including Gomory himself. Here are a few representative quotes reflecting the general sentiment in the early 90's.

In 1991, Gomory (1991) remembered his experience with fractional cuts as follows: "In the summer of 1959, I joined IBM research and was able to compute in earnest... We started to experience the unpredictability of the computational results rather steadily."

G. Cornuéjols (✉)
Tepper School of Business, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213-3890;
Laboratoire d'Informatique Fondamentale, Faculté des Sciences de Luminy, Marseille, France
e-mail: gc0v@andrew.cmu.edu

In 1991, Padberg and Rinaldi (1991) made the following statements about Gomory cuts: "These cutting planes have poor convergence properties... classical cutting planes furnish weak cuts". "A marriage of classical cutting planes and tree search is out of the question as far as the solution of large-scale combinatorial optimization problems is concerned."

In 1989, Nemhauser and Wolsey (1989) had this to say about Gomory cuts: "They do not work well in practice. They fail because an extremely large number of these cuts frequently are required for convergence."

Other textbook authors of the 80's held similar views. Williams (1985) says: "Although cutting plane methods may appear mathematically fairly elegant, they have not proved very successful on large problems." Parker and Rardin (1988) give the following explanation for this lack of success: "The main difficulty has come, not from the number of iterations, but from numerical errors in computer arithmetic."

In the late 80's and early 90's, Sebastian Ceria, Egon Balas and I had great fun investigating lift-and-project cuts (Balas, Ceria, and Cornuéjols, 1993) for 0,1 MILP's. For several instances considered difficult in the early 90's, we were pleased to observe the remarkable strength of these cuts. It went against the conventional wisdom according to which strong cuts had to be designed specifically to exploit the problem structure. Sebastian Ceria wrote a branch-and-cut code based on lift-and-project cuts that was faster and more robust than the best codes available at the time (CplexMIP 2.1, MINTO 1.4 and OSL) (Balas, Ceria, and Cornuéjols, 1996). What do lift-and-project cuts have to do with Gomory cuts? Theorem 3.3 in Balas, Ceria, and Cornuéjols (1993) states that the Gomory mixed integer cuts generated from the rows of an optimal tableau are dominated by lift-and-project cuts. In view of this theorem and the earlier quotes from the literature, Gomory cuts were not appealing. Nevertheless I kept nagging Sebastian, trying to convince him to compare them with lift-and-project cuts: Although Gomory cuts are weaker, they are much faster to compute.

The computational results were striking. The same good results that we had obtained with lift-and-project cuts were also obtained with Gomory cuts. Branch and cut based on Gomory cuts was faster and more robust than pure branch and bound. This went completely against the conventional wisdom.

Here is how Sebastian Ceria remembers it:

"While I was a PhD student, Gerard always insisted that we compare Lift-and-Project cuts with Gomory Cuts. He argued that since Gomory Mixed-Integer Cuts were the only "generic" cuts in the literature, it was the only fair test for Lift-and-Projects cuts. I remember him telling me how disappointed he was to find that there were no comprehensive studies of Gomory Cuts that had been published in the literature, as he frequently used to say, "Nobody likes to publish negative results, how can we really know that a comprehensive study was ever done?". "Well, I am sure somebody tried..."

As it frequently happens when you are a PhD student, I had no choice, I had to do a new computational study of Gomory Cuts. It was the only way to be sure. The first implementation of Mixed-Integer Gomory Cuts was quite challenging. In order to obtain the basis inverse, we had to rely on advanced features of CPLEX that were poorly documented, and it was almost impossible to know a priori what was the right way of calling these routines. I had to do a lot of trial and error, setting up simple examples, retrieving the basis, and comparing the results with what I obtained from LINDO or what I could do by hand by pivoting on a tableau. I also relied on CPLEX's tech support, they were great in trying to help me figure it out. But just when I thought I had a successful implementation, I would invariably find an example in MIPLIB where I would be generating invalid cuts (this was indeed a must-do test!). I remember that my final implementation was quite compact and also somewhat indecipherable, but it worked.

And then, there it was, in the summer of 1993, I had my first successful implementation of Mixed-Integer Gomory cuts, and it was working... The initial results were amazing, quite hard to believe. Gerard was quite excited when we discussed them for the first time. "I told you!" He said full of joy, "I told you they would work!". For the next nine months we worked on refining the strategies, in a completely secretive mode. We wanted to present the results in a special meeting, and we wanted these results to come out as a complete surprise to the audience. After some serious planning, we decided that the best venue for presenting our research would be the conference that Dennis Naddef was organizing at Giens, in May of 1994. But it was very important not to give out any clues! Fortunately, this was easy... I just had to submit an abstract about "general cuts for mixed 0–1 programs" and the title of the presentation was "Branch and Cut with General Cutting Planes"; everybody in the audience would expect that I was going to talk about lift-and-project cuts.

It was also important to build the suspense, so I started the presentation by talking about generic qualities that cutting planes would have, and how to use them in a branch-and-cut framework. But "la piece de resistance" was to show a table of computational results before saying which cuts had been used to obtain them. And then, the surprise: "We use Gomory's Mixed-Integer Cuts". I remember the room vividly; George Nemhauser, Bob Bixby, Martin Grötschel, Egon Balas, Laurence Wolsey, Denis Naddef, Antonio Sassano, Giovanni Rinaldi, Michele Conforti, Jack Edmonds were in the room. First silence, then surprise, and finally, "How can this be possible?". It was the best presentation of my career, with a healthy dose of drama and all. Gerard was proud, and I was happy to make him proud. It was he who made this possible. Who knows? Would commercial mixed-integer codes be using Gomory Cuts if Gerard had not insisted on us trying them out again?"

This is kind of you, Sebastian, but someone would have tried them eventually. In any case, it is puzzling that thirty years elapsed before researchers realized how powerful Gomory cuts are in practice. Our computational experience with MIPLIB instances was striking: By using Gomory cuts in a branch-and-cut framework, we could solve 86% of the instances versus only 55% with pure branch and bound. For those instances that could be solved by both algorithms, the version that used Gomory cuts was faster on average, in a couple of cases by a factor of 10 or more. The details of this experiment can be found in Table 1 of (Balas et al., 1996). This was a big surprise to many in the integer programming community and it took several more years to be accepted.

Why did Gomory cuts work for us when they had not worked well for previous researchers? The truth of the matter is that, although many authors commented on how bad Gomory cuts worked in practice, very few had actually tried them themselves. Those who tried them made poor implementation decisions, such as (i) adopting a pure cutting plane approach and (ii) adding cuts one at a time, reoptimizing the LP after each cut addition. There were three main reasons why Gomory cuts worked for us:

- LP solvers were more robust.
- We added *all* the cuts from the optimal LP tableau (instead of just one cut, as tried by Gomory).
- We used a branch-and-cut framework (instead of a pure cutting plane approach).

Padberg and Rinaldi (1991) voiced reservations about using Gomory cuts within branch and cut. They noted that, when variables have been fixed at a node of the search tree, "Gomory cuts need not be valid for the original integer polyhedron, which makes it necessary to treat and store separately the cuts related to each node. This requires additional bookkeeping and a much larger amount of memory." Egon Balas, Sebastian Ceria and I considered the issue of sharing cuts across branches of the enumeration tree and we proved that, in fact, Gomory cuts can be shared across branches for a mixed 0,1 program. This led to the paper (Balas et al., 1996), after a few more experiments performed by Natraj. Interestingly, we had trouble getting the paper accepted for publication: One referee commented that “there is nothing new” while another was so suspicious of the results that he requested a copy of the code. The associate editor recommended rejection, the editor overruled the decision and eventually the paper got published!

Acknowledgements This work was supported in part by NSF grant DMI-0352885, ONR grant N00014-97-1-0196 and ANR grant BLAN06-1-138894.

References

Balas, E., S. Ceria, and G. Cornuéjols. (1993). “A Lift-and-Project Cutting Plane Algorithm for Mixed 0–1 Programs, *Math. Programming*, 58, 295–324.

Balas, E., S. Ceria, and G. Cornuéjols. (1996). “Mixed 0–1 Programming by Lift-and-Project in a Branch-and-Cut Framework.” *Management Science*, 42, 1229–1246.

Balas, E., S. Ceria, G. Cornuéjols, and N. Natraj. (1996). “Gomory Cuts Revisited.” *Operations Research Letters*, 19, 1–9.

Gomory, R. (1960). *An Algorithm for the Mixed Integer Problem*. Technical Report RM-2597, The Rand Corporation.

Gomory, R. (1963). “An Algorithm for Integer Solutions to Linear Programs.” In R.L. Graves and P. Wolfe (Eds.) *Recent Advances in Mathematical Programming*, pp. 269–302. McGraw-Hill, New York.

Gomory, R. (1991). “Early Integer Programming, In J.K. Lenstra, A.H.G. Rinnooy Kan, and A. Schrijver (Eds.). *History of Mathematical Programming, A Collection of Personal Reminiscences*, pp. 55–61. North-Holland, Amsterdam.

Nemhauser, G.L. and L.A. Wolsey. (1989). “Integer Programming.” In G.L. Nemhauser, A.H.G. Rinnooy Kan and M.J. Todd (Eds.). *Handbook in Operations Research and Management Science 1: Optimization*, pp. 447–527. North-Holland, Amsterdam.

Padberg, M. and G. Rinaldi. (1991). “A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems, *SIAM Review*, 33, 60–100.

Parker, R.G. and R.L. Rardin. (1988). *Discrete Optimization*. Academic Press, New York.

Williams, H.P. (1985). *Model Building in Mathematical Programming*. Wiley, New York.