The Ongoing Story of Gomory Cuts

Gérard Cornuéjols

2010 Mathematics Subject Classification: 90C10, 90C11
Keywords and Phrases: Gomory cut, GMI cut

The story of Gomory cuts is characterized by swings between great acclaim in the early days, near oblivion for decades and an amazing come back in the last 20 years. These cuts have been described as "elegant", "disappointing" and "the clear winner" at various times over the last 55 years. This essay retraces that roller coaster.

Ralph Gomory's paper "Early Integer Programming" recounts his discovery of fractional cuts. It is a few years after he wrote his doctoral dissertation on nonlinear differential equations that he heard of linear programming for the first time. He was working for the Navy at the time. In one particular instance, it would have been preferable to have solutions in integers. Gomory thought that, somehow, one should be able to accomplish this. Within a few days he had invented fractional cuts. His approach was to first solve the linear program and then, using appropriate integer linear forms, to generate valid linear inequalities cutting off the undesirable fractional solution. By adding these cuts to the linear program, solving again using the simplex algorithm and iterating, Gomory could solve by hand any small integer linear program that he tried. However, he did not have a finiteness proof yet. At this point, he happened to run into Martin Beale in the halls of Princeton University in late 1957 and mentioned that he could solve linear programs in integers. When Beale immediately responded "but that's impossible", Gomory realized that he was not the first to think about this problem. As it turns out, Dantzig, Fulkerson, and Johnson had pioneered the cutting plane approach in a seminal paper published in 1954. They devised special-purpose cuts for the traveling salesman problem and, as a result, were able to solve to optimality an instance with 48 cities. However, Gomory's goal was different and more ambitious. His fractional cuts were general-purpose cuts that applied to all integer linear programs. In his reminiscences "Early Integer Programming", Gomory recounts the excitement that followed his encounter with Beale.

> During the exciting weeks that followed, I finally worked out a finiteness proof and then programmed the algorithm on the E101, a pin board computer that was busy during the day but that I could use

> late at night. The E101 had only about 100 characters of memory and the board held only 120 instructions at a time, so that I had to change boards after each simplex maximization cycle and put it in a new board that generated the cut, and then put the old board back to remaximize. It was also hard work to get the simplex method down to 120 E101 instructions. But the results were better and more reliable than my hand calculations, and I was able to steadily and rapidly produce solutions to four- and five-variable problems.

When Gomory presented his results in early 1958, the impact was enormous and immediate. Gomory had achieved the impossible: reducing integer linear programming to a sequence of linear programs. This was a great theoretical breakthrough. The next logical step was to try turning this work into a practical algorithm. In the summer of 1958, Gomory programmed his fractional cutting plane algorithm in FORTRAN (a new computer language at the time). He says

> Most of the problems ran quickly but one went on and on ... it was the first hint of the computational problems that lay ahead ... 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.

In 1960, Gomory [6] extended his approach to mixed-integer linear programs (MILPs), inventing the "mixed-integer cuts", known today as GMI cuts (the acronym stands for Gomory mixed-integer cuts). GMI cuts are remarkable on at least two counts: 1) They are stronger than the fractional cuts when applied to pure integer programs; 2) They apply to MILPs, a crucial feature when generating cutting planes in an iterative fashion because pure integer programs typically turn into MILPs after adding cuts. Three years later, in 1963, Gomory [7] states that these cuts are "almost completely computationally untested." Surprisingly, Gomory does not even mention GMI cuts in his reminiscences in 1991.

In the three decades from 1963 to 1993, Gomory cuts were considered impractical. Several quotes from the late 80s and early 90s illustrate this widely held view. Williams [11]: "Although cutting plane methods may appear mathematically fairly elegant, they have not proved very successful on large problems." Nemhauser and Wolsey [9]: "They do not work well in practice. They fail because an extremely large number of these cuts frequently are required for convergence." Padberg and Rinaldi [10]:

> 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.

By contrast, the Dantzig, Fulkerson, Johnson strategy of generating special-purpose cuts had gained momentum by the early 90s. Padberg and Rinaldi [10] obtained spectacular results for the traveling salesman problem using this approach. It was applied with a varying degree of success to numerous other classes of problems. The effectiveness of such branch-and-cut algorithms was attributed to the use of facets of the integer polyhedron.

Was this view of cutting planes justified? Despite the bad press Gomory cuts had in the research community and in textbooks, there was scant evidence in the literature to justify this negative attitude. Gomory's quote from thirty years earlier was still current: GMI cuts were "almost completely computationally untested." In 1993 I convinced Sebastian Ceria, who was a PhD student at Carnegie Mellon University at the time, to experiment with GMI cuts. The computational results that he obtained on MIPLIB instances were stunning [1]: By incorporating GMI cuts in a branch-and-cut framework, he 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 GMI cuts was faster on average, in a couple of cases by a factor of 10 or more. This was a big surprise to many in the integer programming community and several years passed before it was accepted. In fact, publishing the paper reporting these results, which so strongly contradicted the commonly held views at the time, was an uphill battle (one referee commented "there is nothing new" and requested that we add a theoretical section, another so distrusted the results that he asked to see a copy of the code. The associate editor recommended rejection, but in the end the editor overruled the decision, and the paper [1] was published in 1996).

Our implementation of Gomory cuts was successful for three main reasons:

- We added *all* the cuts from the optimal LP tableau (instead of just one cut, as Gomory did).
- We used a branch-and-cut framework (instead of a pure cutting plane approach).
- LP solvers were more stable by the early 1990s.

Commercial solvers for MILPs, such as Cplex, started incorporating GMI cuts in 1999. Other cutting planes were implemented as well and solvers became orders of magnitude faster. Bixby, Fenelon, Gu, Rothberg and Wunderling [3] give a fascinating account of the evolution of the Cplex solver. They view 1999 as the transition year from the "old generation" of Cplex to the "new generation". Their paper lists some key features of a 2002 "new generation" solver and compares the speedup in computing time achieved by enabling one feature versus disabling it, while keeping everything else unchanged. The table below summarizes average speedups obtained for each feature on a set of 106 instances.

Feature	Speedup factor
Cuts	54
Presolve	11
Variable selection	3
Heuristics	1.5

The clear winner in these tests was cutting planes. In 2002 Cplex implemented eight types of cutting planes. Which were the most useful? In a similar experiment disabling only one of the cut generators at a time, Bixby, Fenelon, Gu, Rothberg and Wunderling obtained the following degradation in computing time.

Cut type	Factor
GMI	2.5
MIR	1.8
Knapsack cover	1.4
Flow cover	1.2
Implied bounds	1.2
Path	1.04
Clique	1.02
GUB cover	1.02

Even when all the other cutting planes are used in Cplex (2002 version), the addition of Gomory cuts by itself produces a solver that is 2.5 times faster! As Bixby and his co-authors conclude "Gomory cuts are the clear winner by this measure". Interestingly the MIR (Mixed Integer Rounding) cuts, which come out second in this comparison, turn out to be another form of GMI cuts!

However, that's not the end of the story of Gomory cuts. More work is needed on how to generate "safe" Gomory cuts: The textbook formula for generating these cuts is not used directly in open-source and commercial software due to the limited numerical precision in the computations; solvers implement additional steps in an attempt to avoid generating invalid cuts. Despite these steps, practitioners are well aware that the optimal solution is cut off once in a while. More research is needed. Another issue that has attracted attention but still needs further investigation is the choice of the equations used to generate GMI cuts: Gomory proposed to generate cuts from the rows of the optimal simplex tableau but other equations can also be used. Balas and Saxena [2], and Dash, Günlük and Lodi [4] provide computational evidence that MILP formulations can typically be strengthened very significantly by generating Gomory cuts from a well chosen set of equations. But finding such a good family of equations "efficiently" remains a challenge.

Acknowledgement

This work was supported in part by NSF grant CMMI 1024554 and ONR grant N00014-12-10032.

References

[1] E. Balas, S. Ceria, G. Cornuéjols and N. Natraj, Gomory cuts revisited, *Operations Research Letters 19* (1996) 1–9.

[2] E. Balas and A. Saxena, Optimizing over the split closure, *Mathemtical Programming 113* (2008) 219–240.

[3] R.E. Bixby, M. Fenelon, Z. Gu, Ed Rothberg and R. Wunderling, Mixed-Integer Programming: A Progress Report, in *The Sharpest Cut: The Impact of Manfred Padberg and His Work*, edited by Martin Grötschel, *MPS-SIAM Series on Optimization* (2004) 309–325.

[4] S. Dash, O. Günlük and A. Lodi, On the MIR closure of polyhedra, *12th International IPCO Conference, Ithaca, NY, June 2007*, (M. Fischetti and D.P. Williamson eds.) *LNCS 4513* (2007) 337–351.

[5] R. Gomory, Outline of an Algorithm for Integer Solutions to Linear Programs, *Bulletin of the American Mathematical Society 64* (1958) 275–278.

[6] R. Gomory, An algorithm for the mixed integer problem, Technical Report RM-2597, The Rand Corporation (1960).

[7] R. Gomory, An algorithm for integer solutions to linear programs, in R.L. Graves and P. Wolfe eds., *Recent Advances in Mathematical Programming*, McGraw-Hill, New York (1963) 269–302.

[8] R. Gomory, 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*, North-Holland, Amsterdam (1991) 55–61.

[9] G.L. Nemhauser and L.A. Wolsey, 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*, North-Holland, Amsterdam (1989) 447–527.

[10] M. Padberg and G. Rinaldi, A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems, *SIAM Review 33* (1991) 60–100.

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

Gérard Cornuéjols
Tepper School of Business
Carnegie Mellon University
Pittsburgh
Pennsylvania 15213-3890
USA
gc0v@andrew.cmu.edu