Math 256B: Final Project

June 8, 2017

David Haley

Background

Integer programming is a vast subfield of optimization in which some or all of the optimization parameters must take on values from a discrete set. At first glance, this means that we can optimize problems such as job assignments in which it is not possible to hire half of an employee, although such direct applications are more far reaching than that. More broadly, we can reduce very difficult theoretical problems from number theory and combinatorial optimization to integer programs, so any advance in integer programming will also provide advances in these other fields [1] [2]. Furthermore, there is a reduction from the 2D Ising model used in physics and networking to an integer program, and it is a matter of current debate if integer programs can more efficiently solve instances of the Ising model than D-Wave quantum annealers [3].

As of right now, a commonly used approach for solving integer programs (as used in commercial solvers) is a method called branch-and-cut, in which the program is solved numerically without using the discrete constraints, then through a series of steps it is broken up into smaller subproblems (branching) and/or the existing constraints are refined (cutting). My project proposal focuses on the latter portion: cutting planes.

Gomory-Johnson cut-generating functions

One presently open area of research revolves around how to find such cutting planes. One method, which works by manipulating a single row of the simplex algorithm output, uses Gomory-Johnson cut-generating functions [2]. While the application of these cut-generating functions is straightforward, finding valid cut-generating functions is not trivial. Presently these functions are found by exhaustive computer discovery [4]. Furthermore, the space of minimal cut-generating functions is known to be convex, and so there is a particular focus on trying to find the extreme functions (that is, minimal valid functions that are not convex combinations of others) [5].

At their core, these functions work in the following way. If we know (from one row of the simplex output) that

$$x_i + \sum_{j \in \mathcal{N}} r_j x_j \in f + \mathbb{Z} \qquad \text{and} \qquad x_j \in \mathbb{Z}_+$$

then it is a straightforward (but perhaps not obvious) exercise in algebra to show that

$$\sum_{j \in \mathcal{N}} \{r_j\} x_j \geq f$$

where $\{\cdot\}$ denotes the fractional part of the argument, i.e. $\{x\} := x - \lfloor x \rfloor$.

By doing this, we have created a new inequality based upon the old constraints, and this inequality serves as a cutting plane. By adding this to the constraint set, we remove some of the solutions that a numerical solver

would arrive at that are not valid for the discrete problem. Ideally, we would like to remove as many of the numerical solutions as possible that are not part of the discrete problem (i.e. we are looking for "deep" cuts).

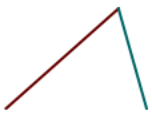
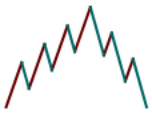
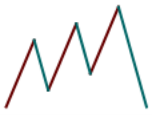
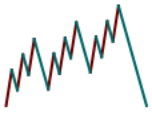
Gomory-Johnson cut-generating functions generalize the role of $\{\cdot\}$ in the above example. Instead, we seek functions $\pi_f(\cdot)$ that can take one row of the simplex tableau and produce a valid cutting plane of this form:

$$\sum_{j \in \mathcal{N}} \pi_f(r_j) x_j \geq \pi_f(f) := 1$$

Such a function π_f is a valid Gomory-Johnson function if this last inequality holds for any feasible x.

The relation between any particular cut-generating function and how deep the cut it will produce is an open problem. This relation is what motivates my project.

Some intution can be gained by looking at some of these cut-generating functions. Computer-assisted discovery has focused on finding such functions that are piecewise linear with rational breakpoints. Some are continuous, but they do not need to be. They vary from simple to exotic. There are catalogues of such functions (I have pasted below a portion of the catalogue from [5]).

`gmic`		2	C	The famous Gomory mixed integer cut, going back to Gomory's 1960 paper [32]. Dominates the `gomory_fractional` cut, which is not minimal (Figure 1).
`gj_2_slope`		2	C	Two families of continuous extreme functions with 2 slopes, from Gomory–Johnson [39]. By the Gomory–Johnson 2-Slope Theorem (Theorem 2.13), all continuous piecewise linear minimal valid functions with 2 slopes are extreme.
`gj_2_slope_repeat`		2	C	
`dg_2_step_mir`		2	C	Described by Dash–Günlük [25]. Extremality follows from the 2-Slope Theorem (Theorem 2.13).
`kf_n_step_mir`		2	C	Described by Kianfar–Fathi [44]. Extremality follows from the 2-Slope Theorem (Theorem 2.13).
`bccz_counterexample`		1–2[b]	C	Limit of `kf_n_step_mir` for $n \to \infty$; not a piecewise linear function. Described by Basu–Conforti-Cornuéjols–Zambelli [12]; see § 6.4

Importantly, most (although not all) are piecewise linear functions with rational breakpoints, and all are maps from $[0, 1)$ to $[0, 1]$. They extend periodically in the domain. What they are qualitatively doing is taking the fractional part of a number in the simplex tableau and changing that into a number in $[0, 1]$.

Project

Each cut-generating function is applied only to the coefficients in one row of the simplex tableau, and so in application the cut-generating function must act in the absence of any knowledge about the coefficients in the other rows. For this reason, in this project I choose to look at the *distribution* of coefficients, and discard the ordering of the coefficients along with their relation to the underlying variables. Viewed in this light, a cut-generating function induces a unique function from one distribution of coefficients to another. To reduce the amount of language needed to describe the remainder of the project, I abuse notation and also refer to this induced function as the cut-generating function π_f, specifying only when the distinction is not clear.

The examplar given earlier in this document is known as the Gomory Fractional Cut ($\pi_f(x) := \{x\}/f$). Being the simplest possible such cut-generating function (that is, the only such function with exactly one linear segment), I use this as a baseline function in order to judge the performance of other cut-generating functions.

For this project I have explored π_f-GFC-neutral distributions, that is, distributions of coefficients that are mapped via π_f to the same distribution of coefficients that would have been produced if the Gomory Fractional Cut had been applied instead. So as not to overload with terminology, I will refer to this as the *scaled invariant distribution*, which I define as the distribution which is invariant under the mapping $f\,\pi_f(\cdot)$.

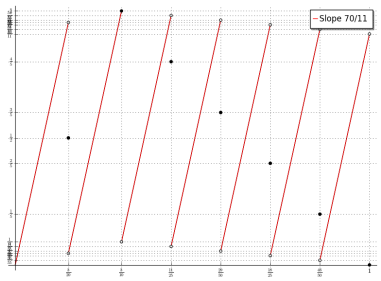
Through the course of exploring the above material through code, the project became an attempt to corroborate a conjecture that if a GJ cut-generating function was applied to coefficients matching the scaled invariant distribution, then the cutting plane produced would not be an improvement over the Gomory Fractional Cut. If this result was provable in closed form, it would generate momentum in the direction of determining how to utilize the (known) distribution of coefficients to intelligently choose which cut-generating function to apply.

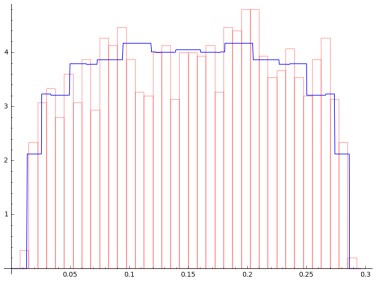
Results

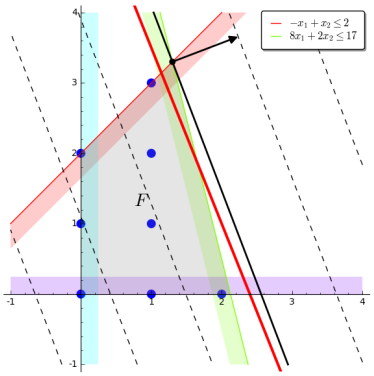
For this project I used Prof. Köppe's Sage code [6] which includes a library of different Gomory-Johnson cut-generating functions, along with parameterized families of these functions which can be instantiated using the provided methods.

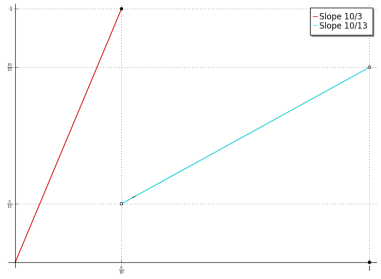
For the experiments I first assume a uniform distribution on $[0, 1]$ (representing total ignorance of the fractional part of the coefficients). By repeated iteration I generated the (scaled) invariant distrution. I then use the function to generate a cut for a simple example and visually assess the deepness of the cut.

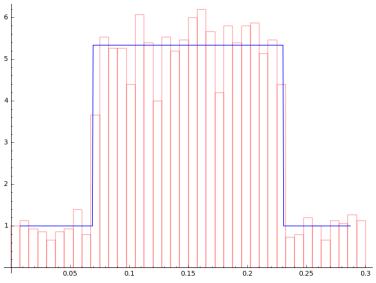
In the following table I give a visual overview of the results of the project. In the first column, the cut-generating function itself is shown. In the second column, the scaled invariant distribution is shown; approximated first by composing the function with itself 19 times (blue line), and then by sampling from a uniform distribution and evolving the sample 197 times (histogram given in red). In the last column, the cut generated by the function on the toy example is shown as a red line overlaid on the feasible region (labelled F). The functions represented in this table are (in order): `drlm_2_slope_limit`, `drlm_3_slope_limit`, `drlm_backward_3_slope`, `gj_2_slope_repeat`, and `ll_strong_fractional`.

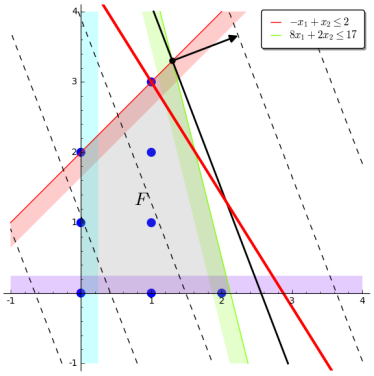

Slope 70/11

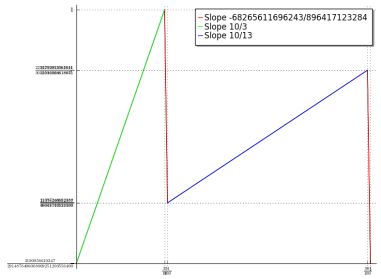


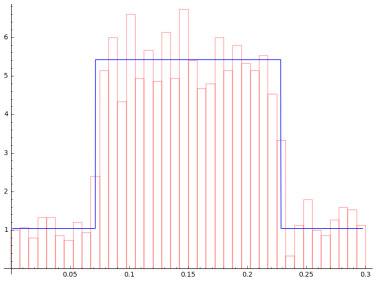

$-x_1 + x_2 \leq 2$
$8x_1 + 2x_2 \leq 17$
F

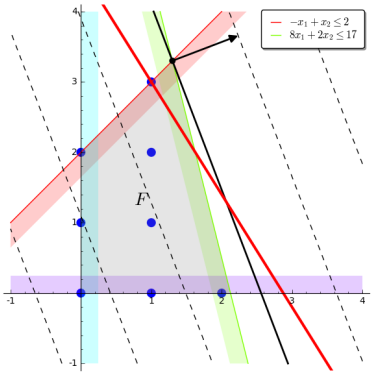

Slope 10/3
Slope 10/13

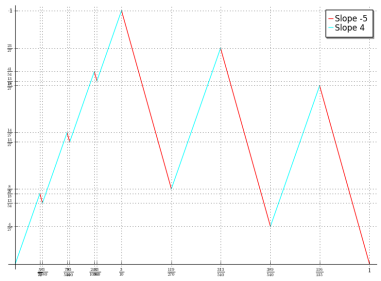


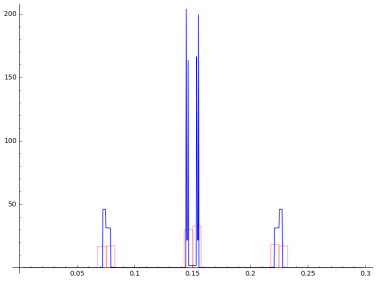

$-x_1 + x_2 \leq 2$
$8x_1 + 2x_2 \leq 17$
F

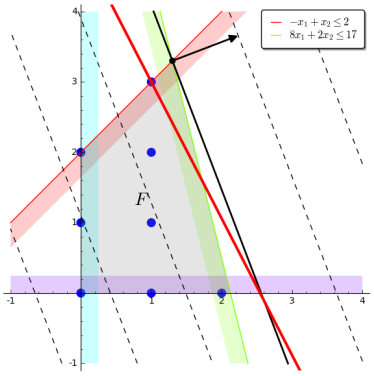

Slope -68265611696243/896417123284
Slope 10/3
Slope 10/13

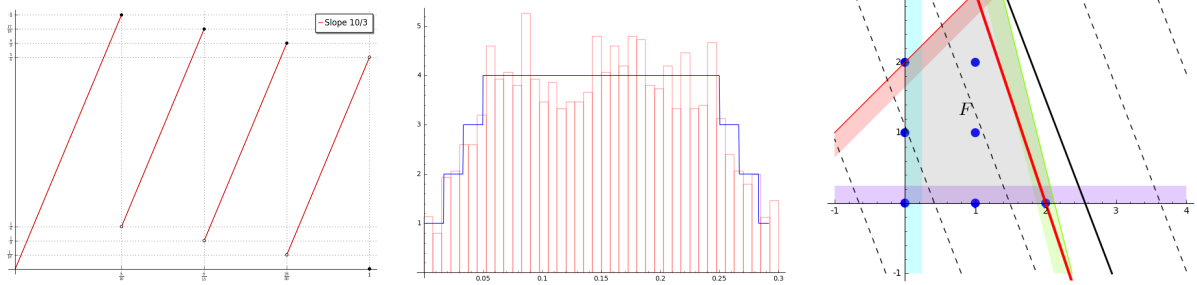



$-x_1 + x_2 \leq 2$
$8x_1 + 2x_2 \leq 17$
F


Slope -5
Slope 4




$-x_1 + x_2 \leq 2$
$8x_1 + 2x_2 \leq 17$
F



Initially it was my hope that there would be some obvious correlation between the invariant distribution and the deepness of the cut, but the results are not conclusive on this point. As such, more experimentation must be done to attempt a definitive result. Among the surprising results is that `ll_strong_fractional` gave the ideal cut; that is, it gave a cutting plane that when added to the constraint set, gives the integral polytope (which can be solved to integrality by one execution of the simplex algorithm). Why this happened for that particular mapping (and why *only* that mapping) is not clear to me at this time.

Future Work

As the results from the first round of experimentation was not promising, I have concluded that I will either need to (a) generate a toy model designed specifically to test the conjecture on a particular mapping, or (b) choose mappings that optimally displace the volume of their invariant distributions away from the coefficients in the toy model. The latter seems to be a more tractable approach, but will require a full examination of the banding behavior of at least one of the maps as it varies by its parameter(s), in much the same manner is done when examining a chaotic map. Once this examination is complete, it will be possible to select a parameter in order to optimize particular banding behavior, and evaluate the performance of this now optimized map on the toy model.

References

[1] Dimitris Bertsimas and Robert Weismantel. *Optimization over integers*, volume 13. Dynamic Ideas Belmont, 2005.

[2] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer programming*, volume 271. Springer, 2014.

[3] Sanjeeb Dash and Jean-Francois Puget. On quadratic unconstrained binary optimization problems defined on chimera graphs. *Mathematical Optimization Society Newsletter*, Optima 98:1–6, 2015.

[4] Yuan Zhou Matthias Köppe. Toward computer-assisted discovery and automated proofs of cutting plane theorems. *ISCO*, 2016.

[5] Amitabh Basu, Robert Hildebrand, and Matthias Köppe. Light on the infinite group relaxation. *arXiv:1410.8584*, 2014.

[6] github.com/mkoeppe/infinite-group-relaxation-code.