Some complexity questions regarding the lazy generation of extremal constraints

David Monniaux

February 18, 2015

Let $S \subseteq \mathbb{Q}^n$. We wish to compute x such that $\forall s \in S \ \langle x, s\rangle \geq 0$.

The set of suitable x is a closed convex cone, the orthogonal of the cone generated by S. In our applications we actually wish x to be in the interior of that cone or, if the interior is empty, in the relative interior. For the sake of discussion, assume now that the cone is full-dimensional, that is, has nonempty interior.

In our application [1] the set S can be expressed so that it is finite, but in general exponentially large and thus we do not wish to ever give it explicitly. We wish to investigate the complexity of finding such an x.

Consider now temporarily our problem as purely a decision problem: is there or not such an x? By Carathéodory's theorem (or is it called a variant of Farkas' lemma?), the solution set is empty if and only if there exists a subset $C \subseteq S$ of size n such that $\{x \mid \forall s \in C \ \langle x, s\rangle \geq 0\}$ is empty. Thus there exists a "small", polynomially-sized witness of that emptiness. This is how in our application we prove coNP membership.

Assume we describe S using a *separation oracle*: given $x_0 \in \mathbb{Q}^n$ the oracle answers whether $\forall s \in S \ \langle x_0, s\rangle \geq 0$ and, if not, provides s_0 such that $\langle x_0, s_0\rangle < 0$. This is sufficient to run the ellipsoid algorithm and obtain a solution x in polynomial time in n and in the bitstring complexity of the elements of S picked by the oracle. (Would we gain anything by going to a *weak separation oracle*? [2, ch. 3])

Thus our intuition is that finding a solution x should not require too many calls to the oracle. In fact, in our applications, the number of iterations is surprisingly small (we have yet to analyze experimental results to see precisely how this grows with the dimension).

Because the practical complexity of the ellipsoid method is very bad, we do not use it and instead use a refinement loop from *extremal* counterexamples. Our algorithm is made a bit complex by the need to account for degenerate cones so here is a simplified version that should work in the case where the solution cone has nonempty interior. Start from an empty set C_0 constraints. At each iteration i, find x_i in the interior of

the cone $\{x \mid \forall s \in C_i \ \langle x, s\rangle \geq 0\}$. Query the oracle whether x_i satisfies $\forall s \in S \ \langle x_i, s\rangle \geq 0$; if so, terminate with x_i as solution. If not the oracle provides s_i such that $\langle x_i, s_i\rangle$ is *minimal* (and < 0), and we set $C_{i+1} = C_i \cup \{s_i\}$ for the next iteration.

Note the difference with a classical oracle approach: instead of just requiring the oracle to supply any separating hyperplane, we require this hyperplane constraint to be saturated by at least one solution x. Thus, we do not iterate over wholly redundant constraints. (Could we still get redundant constraints that all saturate the same extremal x ?)

Is there any known complexity result for such an approach? (We can make the "in the interior" part more precise by specifying how we find it by maximizing some kind of linear barrier function, e.g. the sum of distances to the current constraints.)

References

[1] Laure Gonnord, David Monniaux, and Gabriel Radanne. "Synthesis of ranking functions using extremal counterexamples". In: *Programming Language Design and Implementation (PLDI)*. ACM. 2015.

[2] Levent Tunçel. *Polyhedral and Semidefinite Programming Methods in Combinatorial Optimization*. Fields Institute Monographs. AMS & Fields Institute, 2010. ISBN: 0-8218-3352-9.