| | | |
|---|---|---|
| ORF 523 | PROBLEM SET 6 | Spring 2019, Princeton University |
| Instructor: A.A. Ahmadi | | |
| TA: Cemil Dibek | | Due at 1:30pm, in class, on May 2, 2019 |

**Problem 1: Equivalence of decision and search for some problems in NP**

1. Suppose you had a blackbox that given a 3SAT instance would tell you whether it is satisfiable or not. How can you make polynomially many calls to this blackbox to find a satisfying assignment to any satisfiable instance of 3SAT?

2. Suppose you had a blackbox that given a graph $G$ and an integer $k$ would tell you whether $G$ has a stable set of size larger or equal to $k$. How can you make polynomially many calls to this blackbox to find a maximum stable set of a given graph?

**Problem 2: Complexity of SDP feasibility**

Consider the following decision problem: Given $A_i \in S^{n\times n}, b_i \in \mathbb{R}, i = 1, \ldots, m$, all with rational entries, decide if there exists a matrix $X \in S^{n\times n}$ such that

$$\begin{aligned} &\mathrm{Tr}(A_i X) = b_i, \\ &X \succeq 0. \end{aligned}$$

Determining the complexity of this question is one of the main outstanding open problem in semidefinite programming. At the moment, the problem is not even known to be in NP.[1]

1. One may be tempted to conclude that the problem is in NP because if the SDP is feasible, then we can just write down a solution and check its validity (testing positive semidefiniteness of a given $n \times n$ matrix can be done in $O(n^3)$). Produce a family of SDP feasibility problems where any feasible solution takes an exponential number of bits to write down with respect to the input size. (You don't need to put your SDP family in the "standard form" given above.)

2. Produce a family of SOCP feasibility problems where any feasible solution takes an exponential number of bits to write down with respect to the input size.
*Hint: See if you can make your SDP problems involve only $2 \times 2$ matrices.*

[1] Contrast this with testing LP feasibility, which is in P, and with the fact that we can solve SDPs to arbitrary accuracy in polynomial time.

3. Here is another shockingly simple problem whose complexity is unknown: Given positive integers $a_1, \ldots, a_r, k$, decide if

$$\sqrt{a_1} + \cdots + \sqrt{a_r} \leq k.$$

Show that if SDP feasibility is in NP (resp. P), then this problem is in NP (resp. P).

**Problem 3: Containment among linear/quadratic basic semialgebraic sets**

Consider the following decision problem: Given an $m \times n$ matrix $A$, an $m \times 1$ vector $b$, an $n \times n$ symmetric matrix $P$, an $n \times 1$ vector $q$, and a scalar $r$ (all data is assumed to be rational), test whether

$$\{x \in \mathbb{R}^n |\ Ax \leq b\} \subseteq \{x \in \mathbb{R}^n |\ x^T P x + q^T x + r \leq 0\}.$$

Show that this problem is NP-hard.