University of Konstanz
Department of Mathematics and Statistics
Summer Term 2012

Sebastian Gruler
María López Quijorna
Markus Schweighofer

Polynomial Optimization – Computer Project 2

The aim of this project is to see that SDP relaxations (cf. lecture notes, §1.4) improve dramatically over LP relaxations (cf. lecture notes, §1.3).

You will again need MATLAB and its Symbolic Math Toolbox MuPAD. As last time, you need to have installed and added to the MATLAB path YALMIP, at least one of the SDP-solvers SeDuMi and SDPT3, and at least one of the nine LP solvers (five free and four commercial) supported by YALMIP[1] (an easy way to install all of these is to install MPT[2]).

In this project, you have to construct five files inside a new directory named `pop2narendra` where `narendra`[3] must be replaced by your given name in lowercase letters:

(1) a MuPAD notebook `lpsdp.mn`

(2) a MuPAD program `lpsdp.mu`

(3) a MATLAB function file `lprelax.m`

(4) a MATLAB function file `sdprelax.m`

(5) a MATLAB script `lpsdp.m`

All five files have to contain a comment with your name in the first line. Apart from this, the only files which have to be commented are (1) and (5). Indeed, (1) should contain a well-documented version of (2) whose only aim is to explain (2). The MuPAD file (2) will contain two procedures `lprelax:=proc(f,k)...` and `sdprelax:=proc(f,k)...` constructing an LP and an SDP relaxation, respectively, of the POP

$$(P) \qquad \text{minimize } f(x) \text{ over all } x \in \mathbb{R} \text{ subject to } x \geq 0 \text{ and } 1-x \geq 0 \qquad (f \in \mathbb{R}[X]).$$

The MATLAB files (3) and (4) contain two MATLAB functions `function [lower,upper,guess] = lprelax(f,k)...` and `function [lower,upper,guess] = sdprelax(f,k)...` calling (2) and (3), respectively, and solving the corresponding relaxation. The MATLAB script (5) contains a number of calls of `lprelax` and `sdprelax` with interesting polynomials and degrees of relaxations that you choose according to our instructions below and also according to your own taste and creativity. You should

[1] `http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Solvers.Solvers`
[2] `http://control.ee.ethz.ch/~mpt/`
[3] `http://en.wikipedia.org/wiki/Karmarkar%27s_algorithm`

comment on the outcomes and observations, you might also include plots of the corresponding polynomials on your interval $[0, 1]$ wherever it is instructive.

All files must be executable without producing errors. Note that this must work wherever your directory `pop2narendra` is placed so please avoid using pathnames when specifying filenames. It is perfectly allowed to collaborate with other students. However, the finalization, annotation and submission of the project has to be done by each participant individually. Comments should be concise and in English language.

(a) Write a MuPAD function `lprelax:=proc(f,k)...` with input parameters $f \in \mathbb{R}[X]$ and $k \in \mathbb{N}$ that

- augments the POP (P) by the redundant inequalities

$$x^i(1-x)^j \geq 0 \quad (i, j \in \mathbb{N}_0, i + j \leq k),$$

- relaxes the augmented POP to an LP (L_k) by replacing each monomial X^i with a new variables Y_i as in §1.3 of the lecture notes, and
- writes the (data defining) (L_k) in YALMIP format into a single string `yal` and returns this string. Before returning `yal`, the carriage returns appearing in it should be deleted as in:

```
yal:=stringlib::subs(yal,"zeros"="sdpvar","t0"="1","\n"="");
```

(b) Argue why the LP dual to (L_k) (lecture notes, 2.3.11) can be interpreted as

$$(D_k) \qquad \text{maximize } \mu \text{ over all } \mu \in \mathbb{R} \text{ subject to } f - \mu \in T_k$$

where T_k is defined as in Exercise 2 on Sheet 4 (this exercise is easy and need not be submitted). In the following, keep in mind the strong duality of linear optimization (lecture notes, 2.3.13).

(c) Write a MuPAD function `sdprelax:=proc(f,k)...` with input parameters $f \in \mathbb{R}[X]$ and $k \in \mathbb{N}$ that

- blows up the constraints of the POP (P) to three polynomial matrix inequalities corresponding to the three families of redundant inequalities

$$\begin{aligned}(a_0 + a_1X + \cdots + a_lX^l)^2 \geq 0 &\quad (a_i \in \mathbb{R}, 2l \leq k),\\ X(a_0 + a_1X + \cdots + a_lX^l)^2 \geq 0 &\quad (a_i \in \mathbb{R}, 2l + 1 \leq k),\\ (1 - X)(a_0 + a_1X + \cdots + a_lX^l)^2 \geq 0 &\quad (a_i \in \mathbb{R}, 2l + 1 \leq k),\end{aligned}$$

- relaxes the blown up POP to an SDP (S_k) by replacing each monomial X^i with a new variables Y_i as in §1.4 of the lecture notes, and
- writes (the data defining) (S_k) in YALMIP format into a single string `yal` and returns this string. Before returning `yal`, the carriage returns appearing in it should again be deleted.

(d) Write a MATLAB function `function [lower,upper,guess] = lprelax(f,k)` with input parameters $f \in \mathbb{R}[X]$ and $k \in \mathbb{N}$ that

- defines a symbolic variable `x` by `syms x`,
- reads (2) by calling `read(symengine,'lpsdp.mu')`,
- defines the LP relaxation of degree k by calling `eval(char(feval(symengine,'lprelax',f,k)))`,
- calls YALMIP using `solvesdp` (in order to have better performance make sure that YALMIP calls an LP solver instead of an SDP solver by adding the LP solver you have installed to your MATLAB path),
- returns the computed optimal value L_k^* of (L_k) which is a lower bound of P^*,
- returns the value of f at `double(y(1))` $\in [0,1]$ (using `subs`) which is an upper bound of (P^*) where `y(1)` $= Y_1$ is the variable in (L_k) relaxing the variable $X = X^1$ from (P), and
- returns the value `double(y(1))` $\in [0,1]$ as a guess for an optimal solution $x^* \in [0,1]$ of (P).

(e) Write a MATLAB function `function [lower,upper,guess] = sdprelax(f,k)` analogous to (d).

(f) Write a MATLAB script `lpsdp.m` which can be read as an essay about the different behavior of LP and SDP relaxations of POPs of the form (P). Amongst others, there should be examples illustrating the behavior of the LP relaxations on certain types of polynomials predicted by the different parts of Exercise 2 on Sheet 4, taking into account (b) above. Moreover, there should be examples comparing the results and the performance of the LP and SDP relaxations. Another topic could be to find examples where the guess for the minimizer $x^* \in [0,1]$ is totally wrong even for high degree of relaxation k. Be creative and show us your discoveries about LP and SDP relaxations of POPs!

Due by Thursday, June 14th, 2012, 11:11 am. The five files (1)—(5) must be sent in a folder to `leonid.chatschijan@uni-konstanz.de` where `leonid.chatschijan` must be replaced by your tutor `sebastian.gruler` or `maria.lopez-quijorna`.