University of Konstanz
Department of Mathematics and Statistics
Summer Term 2019

Alexander Taveira Blomenhofer
Markus Schweighofer

Polynomial Optimization – Computer Project 2

General remarks. This project will later be continued by Computer Project 3. The aim of the project is to get a good intuition on the benefit of adding redundant inequalities to the constraints of a polynomial optimization problem. To this end, we consider the example of the combinatorial problem from Exercises 3 on Sheets 2 and 3. We will only add a finite number of very special redundant inequalities before the linearization. In this way, our relaxations will be even LPs instead of SDPs. We can solve them using MOSEK or any other LP solver supported by YALMIP.[1]

We will also implement for the first time a *rounding procedure*[2] that tries to guess a feasible solution of the original POP[3] from a feasible solution of the relaxation.

You will again need the commercial software MATLAB together with its free package YALMIP. What else you will need, depends on how you solve the project. There are essentially three alternative ways you can proceed:

- The **professional but futureless** solution: You can proceed in a similar way as in Computer Project 1 and use the fast and elegant MuPAD language that still is part of MATLAB's Symbolic Math Toolbox although it is announced to be removed form the Symbolic Math Toolbox in the future. The advantage of this option is that you can program a part of your code in a fast and elegant language that you have already learned about in Computer Project 1. The main disadvantage is that the MuPAD language seems to have no future.[4] You can learn more about the decay of MuPAD at the website of Benno Fuchssteiner.[5] Another disadvantage is that you might need to further explore the poor interface between Matlab and MuPAD.

- **Alexander's** solution: The advantage of this solution is that it does not need anything else other than Matlab and YALMIP. Calculations with polynomials are here entirely based on Löfberg's brilliant YALMIP extension of MATLAB which is however a bit of a hack. The disadvantage is that YALMIP is not made for symbolic computation. Therefore, if at any time in your life you want to solve a problem involving large-scale symbolic computation, you might want to use other software.

[1] `https://yalmip.github.io/allsolvers/`
[2] the one from Exercise 3(h) on Sheet 3
[3] the one from Exercise 3(c) on Sheet 2
[4] `https://de.mathworks.com/help/symbolic/getting-started-with-mupad.html`
[5] `http://www.fuchssteiner.ch`

- **Markus'** solution: Here one uses symbolic variables from MATLAB's Symbolic Math Toolbox but not from MuPAD to calculate with polynomials before the linearization. After the linearization one uses however YALMIP's "sdp variables" to model the optimization problems. The advantage of this solution is that one could still hope that MATLAB's Symbolic Toolbox will be further developed in the future (despite MuPAD being disintegrated) so that one day it it could perhaps be an acceptable choice for symbolic calculations with polynomials. The disadvantage is that it is this is the slowest option of all and that you have to deal with both "symbolic variables" and "sdp variables" at the same time.[6][7]

You will have to construct at least the following files inside a new directory named `pop2narendra` where `narendra`[8] must be replaced by your given name in lowercase letters:

(1) a MATLAB function file `list_schedule.m`

(2) a MATLAB script `franzsepp.m`

If you opt for the **professional but futureless** solution or for **Markus'** solution, then you will need other files as well.

- In case of the **professional but futureless** solution you can find out yourself since you were already trained in Computer Project 1. However, you might want to enable better communication between MuPAD and MATLAB by including lines as:

```
read(symengine,'sheraliadams.mu');
eval(char(feval(symengine,'sheraliadams',k,t,R)));
```

- In case of **Alexander**'s solution, you are allowed to use the following "magic" block of code based on YALMIP's `getvariables`[9] and YALMIP's secret function `variablereplace` that linearizes a polynomial or a matrix polynomial:

```
vars = unique(getvariables(P));
y = sdpvar(length(vars), 1);
P = variablereplace(P,vars,getvariables(y));
```

  Hint: Apply this code to a long vector P containing *all* relevant inequalities and equations. You need to understand what it does and comment it. MATLAB's commands `reshape`[10] and `unique`[11] might be helpful.

[6] https://de.mathworks.com/matlabcentral/answers/462405-polynomials-in-symbolic-toolbox-basic-operations-ex s_tid=prof_contriblnk

[7] https://superuser.com/questions/1437491/matlab-symbolic-toolbox-much-slower-than-mupad-i-thought-it-relie

[8] http://en.wikipedia.org/wiki/Karmarkar%27s_algorithm

[9] https://yalmip.github.io/command/getvariables/

[10] https://www.mathworks.com/help/matlab/ref/reshape.html

[11] https://de.mathworks.com/help/matlab/ref/unique.html?searchHighlight=unique&s_tid=doc_srchtitle

- In case of **Markus'** solution, you are allowed to download, to use and to modify the following MATLAB function files:

```
http://www.math.uni-konstanz.de/~schweigh/19/monomials.m
http://www.math.uni-konstanz.de/~schweigh/19/linearize.m
```

If you use these files, you have to comment their code in order to document that you have understood it.

All submitted files have to contain a comment with your name in the first line. All files need to contain sufficiently many comments that help to understand your code. The MATLAB script (2) should be an exciting presentation of your findings. Your tutor should have fun while executing (2)!

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.

Project description. You work as an IT consultant for the Zürich based company "Julia Bär Group AG". Your earn twice the money that Franz and Sepp earn together. The Konstanz based company "Seefraß" where Franz and Sepp work gives you a work contract until June 5. Your task is to implement a good scheduling procedure on which Franz and Sepp can rely on in the future.

(a) Write a MATLAB `function x = list_schedule(compl_time,t,R)`, which starting from a matrix $R \in \{0,1\}^{k \times k}$ and a list `compl_time` $\in \{1,\dots,t\}^k$ of fractional completion times, effects the algorithm from Exercise 3(h) on Sheet 3 until $x \in S_{k,t,R}$ is a schedule as desired or the maximum number t is exceeded. Note that the choice of the renumbering of $\{1,\dots,k\}$ is irrelevant as long as the renumbered jobs satisfy `compl_time`$(i) \leq$ `compl_time`(j) for $i \leq j$. In fact, it is easy to see that you don't even need to implement an explicit renumbering, but can make use of MATLAB's `min` function.

(b) Write a MATLAB script file `franzsepp.m` that is able to build and solve the LP from Exercise 3(d) on Sheet 2. Apply this at least for $t \in \{3,4\}$ to the sample scenario described at the beginning of Exercise 3 on Sheet 2.

(c) Extend `franzsepp.m` so that it extracts from an optimal solution of the LP the corresponding vector `compl_time` of completion times. Call the `list_schedule` algorithm with respect to `compl_time`. Observe what you get in the sample scenario for $t \in \{3,4\}$.

(d) The Seefraß CEO is very pleased by your work. He thinks that the procedure you have now implemented always finds a schedule (for the given amount of time t) if there exists one. However, after some time, he has strong doubts. What do you think?

Hint. Slightly modify the matrix R from the sample scenario so that you can still find by hand a schedule for $t = 3$. Test the the algorithm from Exercise 3(h) on Sheet 3 on it (i.e., solve the LP and apply the `list_schedule` function).

(e) Since the deadline June 5 is approaching, your boss sends the two company's best IT experts Sherali and Adams. During their exchange semester in Konstanz they had attended a lecture on Polynomial Optimization by Doctor Schweighofer. They vaguely remember some ideas of the lecture. On their flight from Dublin to Zürich in the business class they have the following idea: In the POP from Exercise 3(c) on Sheet 2, they add the following redundant constraints:

$$\begin{aligned}(1 - x_{is})\ell(x) &\geq 0\\ x_{is}\ell(x) &\geq 0\end{aligned}$$

for each *linear* inequality $\ell(x) \geq 0$ occurring in the POP and each $i \in \{1, \ldots, k\}$ and $s \in \{1, \ldots, t\}$. They proceed analogously for the *linear* equations $\ell(x) = 0$ occurring in this POP. As for the quadratic constraints $x_{is} = x_{is}^2$, they just keep them but they keep in mind that the redundant inequalities $x_{ju} \geq 0$ for each j and u were part of the formulation of $S^{LP}_{k,t,R}$ and so they multiply them by x_{is} and $(1 - x_{is})$ as well and add them to the constraints. During his time in Konstanz, Sherali often enjoyed the services of Seefraß while Adams swam in the lake. That's why he comes up with the idea of suppressing the original linear inequalities and equalities $\ell(x) \geq 0$ and $\ell(x) = 0$. After a while he can convince Adams that this is a good idea. They linearize the POP originated in this way by introducing new variables (at least for the non-linear monomials that are now all quadratic) to an LP. Any fractional schedule that arises from a feasible solution of this LP (in the sense that it can be completed to a feasible solution of this LP by assigning values to the variables introduced for the non-linear monomials) is called a *Sherali-Adams schedule*. We denote the set of Sherali-Adams schedules by $S^{\mathrm{SA}}_{k,t,R}$. Sherali says that obviously each Sherali-Adam schedule is a fractional schedule, i.e., $S^{\mathrm{SA}}_{k,t,R} \subseteq S^{\mathrm{LP}}_{k,t,R}$. Adam agrees. Do you also agree? Sherali and Adams ask you to extend your code `franzsepp.m` so that their LP is constructed and solved. Verify that, in the sample scenario, $S^{\mathrm{SA}}_{k,4,R} \neq \emptyset$ but $S^{\mathrm{SA}}_{k,3,R} = \emptyset$.

(f) Extend `franzsepp.m` further so that you apply, in a second try, again the `list_schedule` algorithm but this time using the completion times obtained from a Sherali-Adams schedule. What result do you get in the sample scenario?

(g) Try your code in other scenarios, i.e., with different choices of R (you are encouraged to try different values of k and t as well). Do you find an example where

$S^{\mathrm{SA}}_{k,4,R} \neq \emptyset$ but $S_{k,4,R} = \emptyset$? Could it be true that $S^{\mathrm{SA}}_{k,4,R} = S_{k,4,R}$? Can you find an example where the list scheduling algorithm applied to completion times of a Sherali-Adams schedule (i.e., element of $S^{\mathrm{SA}}_{k,t,R}$) does not succeed in finding a schedule (i.e., an element of $S_{k,t,R}$ with the same t)?

Due by Friday, June 5th, 2019 at noon. All files must be sent attached to an electronic mail to Alexander Taveira Blomenhofer [12].

[12] `http://www.math.uni-konstanz.de/~blomenhofer/`