Hybrid SDP Bounding Procedure

Fabio Furini

LAMSADE, Université Paris-Dauphine, 75775 Paris, France

Emiliano Traversi

LIPN, Université Paris 13, 93430 Villetaneuse, France

PGMO 2013, Paris

Research Questions

- Is it worth to use an additional bounding techniques (based on an SDP relaxation) within a Branch&Bound framework?

Research Questions

- Is it worth to use an additional bounding techniques (based on an SDP relaxation) within a Branch&Bound framework?
- Which Branch&Bound framework will profit more from the addition of the SDP bounding techniques?

Research Questions

- Is it worth to use an additional bounding techniques (based on an SDP relaxation) within a Branch&Bound framework?
- Which Branch&Bound framework will profit more from the addition of the SDP bounding techniques?
- Which SDP relaxation has the best trade-off between the time saved by pruning nodes and the time needed for computing the SDP relaxation?

Research Questions

- Is it worth to use an additional bounding techniques (based on an SDP relaxation) within a Branch&Bound framework?
- Which Branch&Bound framework will profit more from the addition of the SDP bounding techniques?
- Which SDP relaxation has the best trade-off between the time saved by pruning nodes and the time needed for computing the SDP relaxation?
- What additional policies are necessary in order to improve the overall computational performances?

Quadratic Formulation (QF) of the BQP

The Quadratic Formulation (QF) of the BQP, with n variables and p constraints, is defined as follows:

$$
\begin{aligned}
\text{(QF)} \quad \max \ & \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} x_i x_j + \sum_{i=1}^{n} L_i x_i \\
& x \in K \\
& x \in \{0,1\}^n,
\end{aligned}
$$

with $Q \in \mathbb{R}^{n \times n}$, $L \in \mathbb{R}^n$, $K = \{x \in \mathbb{R}^n : Ax \geq b\}$, $A \in \mathbb{R}^{p \times n}$ and $b \in \mathbb{R}^p$. Q is a generic symmetric matrix, not restricted to being convex.

Linear Reformulation (LF) of the BQP

Another option for modeling BQP is to linearize the quadratic terms using the RLT inequalities (Sherali and Adams, 1998) and obtain the following Linear Reformulation (LF):

$$
\begin{aligned}
\text{(LF)} \quad \max \; & \sum_{i=1}^{n} \sum_{j=i}^{n} Q_{ij} y_{ij} + \sum_{i=1}^{n} L_i x_i \\
& \left.\begin{aligned}
y_{ij} &\leq x_i \\
y_{ij} &\leq x_j \\
y_{ij} &\geq x_i + x_j - 1 \\
y_{ij} &\geq 0
\end{aligned}\right\} i, j = 1, \ldots, n \\
& x \in K \\
& x \in \{0,1\}^n \, .
\end{aligned}
$$

Quadratic Stable Set Problem (QSSP)

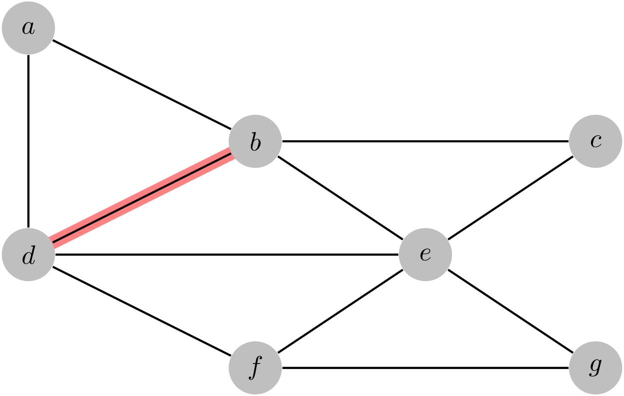
A **Stable** (or **Independent**) Set is a set of vertices in a graph, where no two vertices are adjacent.
In the **maximum-weight Stable Set Problem**, we have an undirected graph with weights on its vertices and we look for an independent set with maximum total weight.

The **maximum-weight Quadratic Stable Set Problem** (QSSP) is a generalization of the classical problem in which we have quadratic costs for couples of vertices. Given an undirected graph $G = (V, E)$:

$$\text{QF of QSSP} \quad \max \sum_{i \in V} \sum_{j \in V} Q_{ij} x_i x_j + \sum_{i \in V} L_i x_i$$
$$x_i + x_j \leq 1 \quad \forall \{i, j\} \in E$$
$$x \in \{0, 1\}^n$$

Quadratic Stable Set Problem (QSSP) - Example

Due to these edge constraints, the only relevant quadratic costs are the ones relative to the couple of nodes that are not connected by an edge:



Quadratic Stable Set Problem (QSSP) - $\tilde{E}$

The only relevant quadratic profits are the ones relative to the couple of nodes that are not connected by an edge.
We hence introduce the complement of E:

$$\tilde{E} = \{\{i,j\} | i,j \in V, i \neq j, \{i,j\} \notin E\}.$$

and hence the QF takes the following form w.l.o.g.:

$$\begin{aligned} \max \sum_{\{i,j\} \in \tilde{E}} & Q_{ij} x_i x_j + \sum_{i=1}^{n} L_i x_i \\ & x \in K \\ & x \in \{0,1\}^n \ . \end{aligned}$$

Quadratic Stable Set Problem (QSSP) - $\tilde{E}$

The only relevant quadratic profits are the ones relative to the couple of nodes that are not connected by an edge.
We hence introduce the complement of E:

$$\tilde{E} = \{\{i,j\} | i,j \in V, i \neq j, \{i,j\} \notin E\}.$$

and hence the QF takes the following form w.l.o.g.:

$$\begin{aligned} \max \sum_{\{i,j\} \in \tilde{E}} & Q_{ij} x_i x_j + \sum_{i=1}^{n} L_i x_i \\ & x \in K \\ & x \in \{0,1\}^n \ . \end{aligned}$$

Moreover, QSSP can be solved as an unconstrained QF by setting $K = \emptyset$ and $Q_{ij} = M$ for all $\{i,j\} \in E$ (M is a suitable big number).

Quadratic Stable Set Problem (QSSP) - $\tilde{E}$

The only relevant quadratic profits are the ones relative to the couple of nodes that are not connected by an edge.
We hence introduce the complement of E:

$$\tilde{E} = \{\{i,j\} | i,j \in V, i \neq j, \{i,j\} \notin E\}.$$

and hence the QF takes the following form w.l.o.g.:

$$\begin{aligned} \max \quad & \sum_{\{i,j\} \in \tilde{E}} Q_{ij} x_i x_j + \sum_{i=1}^{n} L_i x_i \\ & x \in K \\ & x \in \{0,1\}^n \ . \end{aligned}$$

Moreover, QSSP can be solved as an unconstrained QF by setting $K = \emptyset$ and $Q_{ij} = M$ for all $\{i,j\} \in E$ (M is a suitable big number).
(Preliminary tests showed that this approach is not competitive, mainly because of precision problems.)

SDP Reformulation

BQP is equivalent to the following reformulation:

$$\max \langle \tilde{Q}, Y \rangle$$

$$Y = \begin{pmatrix} 1 \\ \bar{x} \end{pmatrix} \begin{pmatrix} 1 \\ \bar{x} \end{pmatrix}^\top \tag{1}$$

$$\bar{x} \in \bar{K}$$

$$\bar{x} \in \{-1, 1\}^n \tag{2}$$

obtained after applying the linear transformation $\bar{x}_i = 2x_i - 1$,

$$\tilde{Q} = \begin{pmatrix} \frac{1}{2}\sum_{j=1}^n L_i + \frac{1}{4}\sum_{i=1}^n \sum_{j=1}^n Q_{ij} & \frac{1}{4}(L + \frac{1}{2}\sum_{j=1}^n Q_j)^\top \\ \frac{1}{4}(L + \frac{1}{2}\sum_{j=1}^n Q_j) & \frac{1}{4}Q \end{pmatrix}$$

with Q_j the j-th column of Q and modifying K in $\bar{K}$ accordingly.

SDP Reformulation and Relaxation

Constraints (1) and (2) can be rewritten as:

$$\max \langle \tilde{Q}, Y \rangle$$
$$Y \succeq 0$$
$$rank(Y) = 1$$
$$diag(Y) = e$$
$$Y_0 \in \bar{K}$$

with e being the all-ones vector and Y_0 being the first row of Y without the first element (note that $Y_0 = \bar{x}$). By relaxing the rank constraints we obtain an SDP relaxation.

SDP Reformulation and Relaxation

Constraints (1) and (2) can be rewritten as:

$$\max \langle \tilde{Q}, Y \rangle$$
$$Y \succeq 0$$

$$diag(Y) = e$$
$$Y_0 \in \bar{K}$$

with e being the all-ones vector and Y_0 being the first row of Y without the first element (note that $Y_0 = \bar{x}$). By relaxing the rank constraints we obtain an SDP relaxation.

SDP Reformulation and Relaxation

Constraints (1) and (2) can be rewritten as:

$$\max \langle \tilde{Q}, Y \rangle$$
$$Y \succeq 0$$

$$diag(Y) = e$$
$$Y_0 \in \bar{K}$$

with e being the all-ones vector and Y_0 being the first row of Y without the first element (note that $Y_0 = \bar{x}$). By relaxing the rank constraints we obtain an SDP relaxation.

For the QSSP, instead of $x_i + x_j \leq 1$ *we have*

$$x_i x_j = 0, \forall \{i, j\} \in E \; .$$

SDP Reformulation and Relaxation

Constraints (1) and (2) can be rewritten as:

$$\max \langle \tilde{Q}, Y \rangle$$
$$Y \succeq 0$$

$$diag(Y) = e$$
$$Y_0 \in \bar{K}$$

with e being the all-ones vector and Y_0 being the first row of Y without the first element (note that $Y_0 = \bar{x}$). By relaxing the rank constraints we obtain an SDP relaxation.

For the QSSP, instead of $x_i + x_j \leq 1$ *we have*

$$\frac{1}{4}(\bar{x}_i + 1)(\bar{x}_j + 1) = 0, \forall \{i, j\} \in E\ .$$

SDP Reformulation and Relaxation

Constraints (1) and (2) can be rewritten as:

$$\max \langle \tilde{Q}, Y \rangle$$
$$Y \succeq 0$$

$$diag(Y) = e$$
$$Y_0 \in \bar{K}$$

with e being the all-ones vector and Y_0 being the first row of Y without the first element (note that $Y_0 = \bar{x}$). By relaxing the rank constraints we obtain an SDP relaxation.
For the QSSP, instead of $x_i + x_j \leq 1$ *we have*

$$Y_{ij} + Y_{0i} + Y_{0j} + 1 = 0, \forall \{i, j\} \in E \ .$$

Hybrid Bounding Procedure

MAIN IDEA: mixing two different relaxations as bounding procedure in a Branch&Bound framework.

Hybrid Bounding Procedure

MAIN IDEA: mixing two different relaxations as bounding procedure in a Branch&Bound framework.
Three main ingredients:

Hybrid Bounding Procedure

MAIN IDEA: mixing two different relaxations as bounding procedure in a Branch&Bound framework.

Three main ingredients:

- Branch&Bound Framework.

Hybrid Bounding Procedure

MAIN IDEA: mixing two different relaxations as bounding procedure in a Branch&Bound framework.

Three main ingredients:

- Branch&Bound Framework.
- Function Ψ, the additional (SDP) bounding procedure.
  Takes as input Q, L and a partial fixing of the variables and returns a bound (UB^{SDP}) on the original objective function.

Hybrid Bounding Procedure

MAIN IDEA: mixing two different relaxations as bounding procedure in a Branch&Bound framework.

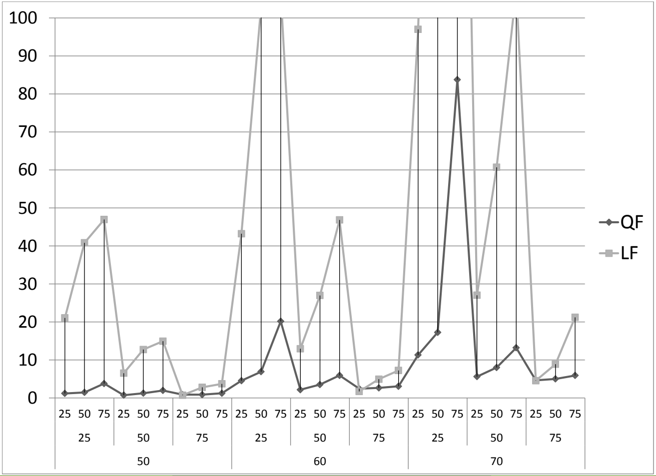
Three main ingredients:

- Branch&Bound Framework.
- Function Ψ, the additional (SDP) bounding procedure. Takes as input Q, L and a partial fixing of the variables and returns a bound (UB^{SDP}) on the original objective function.
- Function Ω, an oracle that takes as input all the information about the current Branch&Bound node and returns a binary variable indicating whether Ψ should be used or not.

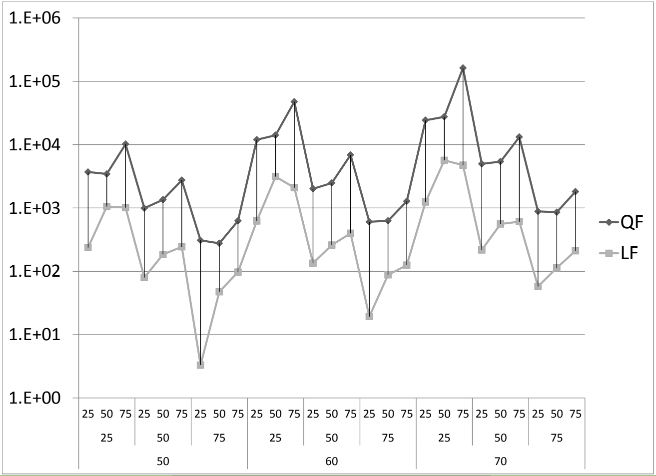
Computational Experiments - Testbed Description

- We randomly generated a set of QSSP instances.
- The instance generator produces random graphs according to the desired number of vertices n and density μ.
- The linear and quadratic profits take a uniformly random integer value in the interval $[-100, 100]$, a third parameter ν represents the percentage of positive profits.
- We generated 27 classes of instances by considering all combinations of:
  - number of vertices: $n \in \{50, 60, 70\}$;
  - density of edges: $\mu \in \{25\%, 50\%, 75\%\}$;
  - percentage of positive costs: $\nu \in \{25\%, 50\%, 75\%\}$.
- We created 10 instances for each class using different random seeds, thus obtaining in total 270 QSSP instances.

Formulation Comparison - B&B time



Formulation Comparison - B&B nodes



Formulation Comparison

- As far as the computing time is concerned, the best mathematical formulation is always QF.
- QF performs a number of nodes on average which is at least one order of magnitude bigger than LF.

The fact that QF explores a larger amount of nodes makes it a more promising candidate for testing the addition of a second bounding procedure.

Formulation Comparison

- As far as the computing time is concerned, the best mathematical formulation is always QF.
- QF performs a number of nodes on average which is at least one order of magnitude bigger than LF.

The fact that QF explores a larger amount of nodes makes it a more promising candidate for testing the addition of a second bounding procedure.
For these reasons, in the rest of the tests we will focus only on QF.

Different SDP strategies Ψ

We propose the following three different options for Ψ:

- `Unconstrained`: SDP relaxation without constraints.
- `Constrained`: SDP relaxation with the family of constraints

$$x_i + x_j \leq 1, \forall \{i, j\} \in E \ .$$

- `Constrained2`: SDP relaxation with the family of constraints

$$Y_{ij} + Y_{0i} + Y_{0j} + 1 = 0, \forall \{i, j\} \in E \ .$$

Different SDP strategies Ψ - Root node

	Ratios						Times					
n	QF	LF	LF$_c$	Uncon	Con1	Con2	QF	LF	LF$_c$	Uncon	Con1	Con2
50	4.0	8.9	2.1	7.5	4.8	1.1	0.1	0.0	7.5	0.0	1.8	1.3
60	4.8	11.8	2.6	9.7	6.0	1.1	0.2	0.1	14.7	0.0	4.6	3.1
70	5.2	14.3	2.9	11.1	6.9	1.1	0.2	0.2	29.2	0.1	10.0	6.9
	4.7	11.7	2.5	9.4	5.9	1.1	0.2	0.1	17.1	0.0	5.5	3.8

Different SDP strategies Ψ - Root node

	Ratios						Times					
n	QF	LF	LF_c	Uncon	Con1	Con2	QF	LF	LF_c	Uncon	Con1	Con2
50	4.0	8.9	2.1	7.5	4.8	1.1	0.1	0.0	7.5	0.0	1.8	1.3
60	4.8	11.8	2.6	9.7	6.0	1.1	0.2	0.1	14.7	0.0	4.6	3.1
70	5.2	14.3	2.9	11.1	6.9	1.1	0.2	0.2	29.2	0.1	10.0	6.9
	4.7	11.7	2.5	9.4	5.9	1.1	0.2	0.1	17.1	0.0	5.5	3.8

- `Constrained1` is dominated by `Constrained2`.

Different SDP strategies Ψ - Root node

	Ratios						Times					
n	QF	LF	LF_c	Uncon	Con1	Con2	QF	LF	LF_c	Uncon	Con1	Con2
50	4.0	8.9	2.1	7.5	4.8	1.1	0.1	0.0	7.5	0.0	1.8	1.3
60	4.8	11.8	2.6	9.7	6.0	1.1	0.2	0.1	14.7	0.0	4.6	3.1
70	5.2	14.3	2.9	11.1	6.9	1.1	0.2	0.2	29.2	0.1	10.0	6.9
	4.7	11.7	2.5	9.4	5.9	1.1	0.2	0.1	17.1	0.0	5.5	3.8

- `Constrained1` is dominated by `Constrained2`.
- Interesting trade-off between `Constrained2` and `Unconstrained`.

Strategies for the Oracle (Ω)

- 1 `Always`. The SDP relaxation Ψ is triggered at each node of the B&B tree.
- 2 `OnOne`. Ψ is triggered every time we branch on one.
- 3 `UnderAverage`. Ψ is triggered if the current integrality gap is lower than the average integrality gap of the nodes explored so far.
- 4 `OverAverage`. Ψ is triggered if the current integrality gap is bigger than the average.
- 5 `SmallGap`. Ψ is triggered only when the integrality gap is within $[0\%, 5\%]$.
- 6 `MediumGap`. Ψ is triggered only when the integrality gap is within $[5\%, 30\%]$.
- 7 `Random`. Ψ is triggered with a random 50% probability.

Strategies for the Oracle (Ω)

			B&B nodes							B&B time						
strategy			*1*	*2*	*3*	*4*	*5*	*6*	*7*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
n	μ	ν	SDP Unconstrained													
70	*25*	*25*	1.5	8.3	1.6	9.6	83.7	20.4	4.7	**78.6**	**65.3**	**65.9**	**63.4**	144.4	**99.5**	**61.3**
		50	13.8	31.9	13.9	57.9	80.4	19.2	29.5	201.3	125.8	149.2	185.9	134.9	113.6	153.7
		75	21.9	43.0	22.8	83.6	79.4	29.2	44.7	227.3	131.2	159.3	201.9	124.1	136.8	177.8
	50	*25*	4.0	15.4	4.1	17.6	96.4	50.7	9.9	**96.9**	**68.1**	**82.1**	**93.9**	102.5	**88.2**	**83.5**
		50	13.6	31.7	14.8	46.6	92.3	43.5	28.1	108.9	**80.2**	**89.4**	111.9	100.5	**86.6**	**97.1**
		75	16.4	36.7	16.9	56.3	90.0	47.2	33.4	105.0	**74.2**	**80.2**	116.2	**97.8**	**80.5**	**93.3**
	75	*25*	10.0	23.3	14.3	31.1	97.1	63.0	20.7	111.6	**90.7**	**95.7**	113.7	**99.9**	**96.3**	103.0
		50	20.8	40.5	22.2	55.8	98.5	65.5	37.9	111.3	**94.9**	**97.1**	113.8	100.3	**97.9**	104.8
		75	18.7	40.0	20.6	57.2	94.6	66.6	36.5	109.6	**89.4**	**91.3**	114.3	**99.5**	**95.7**	101.7
		avg.	13.4	30.1	14.6	46.2	90.2	45.0	27.3	127.8	91.1	101.1	123.9	111.5	99.5	108.5

Table: QF + SDP Unconstrained (Ψ)

Strategies for the Oracle (Ω)

			B&B nodes							B&B time						
		strategy	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*1*	*2*	*3*	*4*	*5*	*6*	*7*
n	*μ*	*ν*	SDP Constrained2													
70	*25*	*25*	0.4	2.1	0.7	3.8	83.7	20.4	1.1	120.3	**65.8**	109.9	**76.4**	186.9	174.0	**83.4**
		50	0.4	1.8	0.8	4.0	80.4	13.9	0.9	**71.8**	**49.4**	**61.2**	**54.1**	160.5	103.5	**46.8**
		75	0.1	0.4	0.2	1.1	79.9	17.7	0.3	**16.6**	**13.4**	**16.1**	**15.4**	133.4	**76.1**	**13.4**
	50	*25*	1.2	6.4	2.7	5.8	96.4	50.7	3.0	765.7	**95.6**	404.3	623.8	103.4	**94.3**	446.8
		50	0.7	5.2	1.8	4.9	92.3	43.9	2.0	232.5	**82.4**	113.7	262.4	101.6	**91.6**	169.3
		75	0.5	3.1	1.4	3.5	90.0	47.0	1.3	207.0	**55.2**	**97.4**	213.5	**98.1**	**82.0**	146.4
	75	*25*	2.9	15.4	8.7	10.1	97.1	63.0	6.0	1033.3	**94.0**	371.7	1139.7	**99.9**	**96.8**	655.1
		50	3.4	13.9	6.6	14.3	98.5	65.4	6.0	796.6	**91.4**	319.9	660.0	100.4	**98.3**	420.9
		75	1.7	8.6	4.3	8.8	94.6	66.6	5.0	1038.2	**85.9**	398.1	1079.1	**99.6**	**95.8**	654.8
		avg.	1.3	6.3	3.0	6.2	90.3	43.2	2.8	475.8	70.4	210.3	458.3	120.4	101.4	293.0

Table: QF + SDP Constrained2 (Ψ)

Results for instances with 100 nodes

			B&B nodes		B&B time		Gold Std.		SDP time		*avg. time*	
		str	*1*	*2*	*1*	*2*	*1*	*2*	*1*	*2*	QF	QF(Ψ,Ω)
n	μ	ν										
100	*25*	*25*	0.1	0.7	**89.5**	**25.0**	13.9	18.3	99.0	95.5	345.4	85.2
		50	0.1	1.0	**49.1**	**24.0**	11.5	17.9	90.9	84.6	419.2	86.9
		75	0.0	0.2	**5.0**	**3.1**	1.3	2.4	90.1	86.4	4378.0	133.7
	50	*25*	0.3	2.5	679.3	**67.0**	57.9	57.4	97.5	50.3	41.6	27.9
		50	0.2	1.9	489.7	**54.9**	42.7	48.9	96.7	41.5	54.5	30.0
		75	0.2	2.2	274.8	**31.0**	22.9	23.6	95.8	48.8	120.6	37.6
	75	*25*	1.5	7.2	3449.9	**80.9**	172.6	79.7	99.3	9.1	20.3	16.4
		50	1.4	11.3	2490.0	**83.1**	164.6	79.7	98.9	9.2	22.7	18.8
		75	0.9	6.4	2269.8	**72.2**	123.9	69.3	98.2	10.3	27.0	19.5
		avg.	0.5	3.7	1088.5	49.0	67.9	44.1	96.3	48.4	603.3	50.7

Table: QF + Constrained2 SDP (Ψ) - Big Instances

Conclusions

- In this work we have explored the use of SDP-based bounding procedures within the B&B framework provided by the MINLP solvers.
- We performed an extensive computational analysis on the QSSP that allows us to conclude that the Hybrid SDP Bounding Procedure allows an average 50% cut of the overall computing time and a cut of more than one order of magnitude for the branching nodes.
- The SDP bounds help in pruning but are heavy to compute, and thus in this optic we proposed different strategies in order to make the Hybrid bounding procedure more efficient.
- The addition of these strategies is crucial for the improvement of the performances over the standard B&B.

Conclusions

THANK YOU!