Decomposition methods for quadratic programming

Lucas Létocart

LIPN - CNRS - Univ. Sorbonne Paris Nord

ROA
2022

Context

- In this work we aim at investigating a family of decompositions for Quadratic Problems (QPs).
- A generic QP reads as follows:

Quadratic Problem (BQP)

$$(QP) \quad \max\{f(x) = x^\top Qx + L^\top x | Ax \leq b, x \in X\} \ .$$

- $Q \in \mathbb{Q}^{n\times n}$ and not restricted to be convex.
- $L \in \mathbb{Q}^n$.
- $X \subseteq \mathbb{R}^n$ or $X \subseteq \mathbb{N}^n$.

Dantzig-Wolfe decomposition for Binary Quadratic Problems (BQPs) *Alberto Ceselli (Univ. Milano) and Emiliano Traversi (Univ. Paris 13)*

A generic BQP reads as follows:

Binary Quadratic Problem (BQP)

$$(BQP) \quad \max\{f(x) = x^\top Qx + L^\top x | Ax \leq b, x \in \{0,1\}\} .$$

- Let A', A'' and b', b'' be a generic row partition of the constraint matrix A and of the rhs vector b.
- The continuous relaxation of (BQP) can be strengthened by convexifying the constraints $A''x \leq b''$ (i.e. imposing $x \in conv\{A''x \leq b'', x \in \{0,1\}\}$).

Simplicial decomposition for Convex Quadratic Problems (CQPs) *Enrico Bettiol (Univ. Paris 13), Francesco Rinaldi (Univ. Padova), Emiliano Traversi (Univ. Paris 13)*

A generic CQP reads as follows:

Convex Quadratic Problem (CQP)

$$(CQP) \quad \max\{f(x) = x^\top Qx + L^\top x | Ax \leq b, x \in \mathbb{R}^n\} .$$

- $Q \in \mathbb{Q}^{n \times n}$ convex.
- $L \in \mathbb{Q}^n$.
- The problem is decomposed keeping the original objective function in the master and the original constraints in the pricing.

Dantzig-Wolfe reformulation and Completely Positive relaxation for binary QCQPs

Enrico Bettiol (Univ. Paris 13), Immanuel Bomze (Univ. of Vienna), Francesco Rinaldi (Univ. Padova), Emiliano Traversi (Univ. Paris 13)

A generic QCQP reads as follows:

$$\langle M, X \rangle := \text{Tr}(M^\top X).$$

Extended formulation

$$\begin{aligned} & \min \langle Q, X \rangle \\ \text{s. t.} \quad & \langle A_i, X \rangle \leq b_i, \quad \forall i = 1 \ldots, m \\ & X = xx^T \\ & x \in \{0,1\}^n \end{aligned}$$

DWR with quadratic master problem I

$$
\begin{aligned}
(BQP_{DWR(A'')}) \quad & \max \quad f(x) && && (1)\\
& \text{s.t.} \quad A'x \leq b' && && [\alpha] \quad (2)\\
& x_j = \sum_{p \in \mathcal{P}_{DWR(A'')}} x_j^p \lambda^p && j = 1, \dots, n && [\tau_j] \quad (3)\\
& \sum_{p \in \mathcal{P}_{DWR(A'')}} \lambda^p = 1 && && [\beta] \quad (4)\\
& x_j \in \{0, 1\} && j = 1, \dots, n && (5)\\
& \lambda^p \geq 0 && p \in \mathcal{P}_{DWR(A'')} && (6)
\end{aligned}
$$

DWR with quadratic master problem II

if f is convex and with $\mathcal{P}_{DWR(A'')}$ being the set of extreme points of $conv\{x|A''x \leq b'', x \in \{0,1\}\}$.

DWR with quadratic master problem III

Convexification of the objective function

If the objective function is non convex, we need to replace $f(x)$ by an equivalent convex objective function $f'(x)$.

$$
\begin{aligned}
(BQP_{DWR(A'')}) \quad & \max \quad f'(x) && && (7)\\
& \text{s.t.} \quad A'x \leq b' && && [\alpha] \quad (8)\\
& x_j = \sum_{p \in \mathcal{P}_{DWR(A'')}} x_j^p \lambda^p && j = 1, \dots, n && [\tau_j] \quad (9)\\
& \sum_{p \in \mathcal{P}_{DWR(A'')}} \lambda^p = 1 && && [\beta] \quad (10)\\
& x_j \in \{0, 1\} && j = 1, \dots, n && (11)\\
& \lambda^p \geq 0 && p \in \mathcal{P}_{DWR(A'')} && (12)
\end{aligned}
$$

DWR with quadratic master problem IV

- Variables λ are partially enumerated by solving an additional *pricing* problem.

Let α, τ and β being the dual variables associated to the constraints in the continuous relaxation of $(BQP_{DWR(A'')})$.

Pricing problem

$$
\begin{aligned}
(\Pi_{BQP_{DWR(A'')}}(\tau^*, \beta^*)) \quad \max \quad & {\tau^*}^\top x + \beta^* && && (13)\\
\text{s.t.} \quad & A'' x \leq b'' && && (14)\\
& x_j \in \{0, 1\} && j = 1, \dots, n && (15)
\end{aligned}
$$

If the optimal value of $(\Pi_{BQP_{DWR(A'')}}(\tau^*, \beta^*))$ is greater than zero, then a column with positive reduced cost is found and added to the master.

DWR with quadratic master problem V

- $(BQP_{DWR}) \Rightarrow f(x)$ is quadratic, the pricing problem is (binary) linear.

DWR with quadratic pricing problem I

The objective function can be rewritten directly in terms of the λ variables by introducing $f(\lambda) = \sum_{p \in \mathcal{P}_{DWR(A'')}} c_p \lambda_p$ with

$$c_p = f(x_p) = x_p^\top Q x_p + L^\top x_p \ .$$

$\overline{DWR}$ of constraints A''

$$(BQP_{\overline{DWR}(A'')}) \quad \max \sum_{p \in \mathcal{P}_{DWR(A'')}} c_p \lambda_p \tag{16}$$

$$\text{s.t.} \quad (8) - (12)$$

DWR with quadratic pricing problem II

Pricing problem

$$
\begin{aligned}
(\Pi_{BQP_{\overline{DWR}(A'')}}(\tau^*, \beta^*)) \quad \max \quad & x^\top Q x + L^\top x + \tau^{*\top} x + \beta^* && (17)\\
\text{s.t.} \quad & A'' x \leq b'' && (18)\\
& x_j \in \{0,1\} \qquad j = 1, \dots, n && (19)
\end{aligned}
$$

where Q is not required to be convex.

- $(BQP_{\overline{DWR}}) \Rightarrow f(\lambda)$ is linear, the pricing problem is (binary) quadratic.

DWR with quadratic pricing problem III

The objective function can still be modified using the convexified objective function $f'(x)$.
The pricing reduces to the following quadratic problem:

Pricing problem

$$(\Pi_{BQP_{\overline{DWR}(A'')}}(\tau^*, \beta^*)) \quad \max \quad f'(x) + {\tau^*}^\top x + \beta^* \tag{20}$$

$$\text{s.t.} \quad A''x \leq b'' \tag{21}$$

$$x_j \in \{0, 1\} \qquad j = 1, \ldots, n \tag{22}$$

(kQKP) Formulation I

Notations

n : number of items
a_j : weight of item j ($j = 1, \ldots, n$)
b : capacity of the knapsack
c_{ij} : profit associated with the selection of items i and j ($i, j = 1, \ldots, n$)
k : number of items to be filled in the knapsack

Assumptions

$c_{ij} \in \mathbb{N}\ i, j = 1, \ldots, n,\ a_j \in \mathbb{N}\ j = 1, \ldots, n,\ b \in \mathbb{N}$
$\max_{j=1,\ldots,n} a_j \leq b < \sum_{j=1}^{n} a_j$
$k \in \{1, \ldots, k_{max}\}$

(kQKP) Formulation II

Mathematical formulation

$$
(\mathrm{kQKP}) \begin{cases} \max \ f(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_i x_j \\ \text{s.t.} \quad \sum_{j=1}^{n} a_j x_j \leq b \quad (1) \\ \sum_{j=1}^{n} x_j = k \quad (2) \\ x_j \in \{0,1\} \quad j = 1, \ldots, n \end{cases}
$$

- without constraint (2): the 0-1 quadratic knapsack problem (QKP)
- without constraint (1): the k-cluster problem

Reformulations for (kQKP)

Reformulation of the objective function $f(x)$

QCR/MIQCR Method

($f(x)$ can be reformulated by exploiting the property $x^2 = x$ and the constraints)

$\Downarrow$

convex problem

Reformulation of the feasible region $\{x|Ax \leq b, x \in \{0,1\}\}$

Dantzig-Wolfe Reformulation

(a subset of constraints is substituted by its convex hull)

$\Downarrow$

tighter formulation

The application of the QCR method leads to the following reformulation of $(kQKP)$:

$$
\begin{aligned}
(kQKP^{\text{conv}}) \quad \max \quad & f_{u,v}(x) && (23)\\
\text{s.t.} \quad & \sum_{j=1}^{n} a_j x_j \leq b && (24)\\
& \sum_{j=1}^{n} x_j = k && (25)\\
& x_j \in \{0,1\} \qquad j = 1,\ldots,n && (26)
\end{aligned}
$$

with

$$
f_{u,v}(x) = f(x) - \sum_{i=1}^{n} u_i \left(x_i^2 - x_i\right) - v \left(\sum_{j=1}^{n} x_j - k\right)^2 \qquad (27)
$$

The application of the MIQCR method leads to the following reformulation of $(kQKP)$:

$$
\begin{aligned}
(kQKP^{\text{impr. conv}}) \quad \max \quad & f_{u,v,P,N}(x,y) && && (28)\\
\text{s.t.} \quad & \sum_{j=1}^{n} a_j x_j \le b && && (29)\\
& \sum_{j=1}^{n} x_j = k && && (30)\\
& y_{ij} \le x_i,\ \ y_{ij} \le x_j && i,j = 1,\dots,n && (31)\\
& y_{ij} \ge 0,\ \ y_{ij} \ge x_i + x_j - 1 && i,j = 1,\dots,n && (32)\\
& x_j \in \{0,1\} && j = 1,\dots,n && (33)
\end{aligned}
$$

with

$$
f_{u,v,P,N}(x,y) = x^T(C - Diag(u) - P - N)x + u^T x + \sum_{i,j=1}^{n}(P_{ij} + N_{ij})y_{ij} - v\left(\sum_{j=1}^{n} x_j - k\right)^2
$$

DWR with a quadratic master problem

- PRO : linear pricing problem.
- CON : the objective function must be convex. $\Rightarrow$ DWR must be applied to $(kQKP^{\text{conv}})$ or to $(kQKP^{\text{impr. conv}})$.

By applying one of the two convexification methods, we always obtain a convex quadratic (binary) optimization problem, whose objective function is of the form:

$$\max \quad \sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{q}_{ij} x_i x_j + \sum_{j=1}^{n} \tilde{l}_j x_j + \sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{w}_{ij} y_{ij}.$$

Reformulation of ($kQKP^{\text{conv}}$) I

$$
\begin{aligned}
(kQKP^{\text{conv}}_{\text{DWD}(\Omega)}) \max \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{q}_{ij} x_i x_j + \sum_{j=1}^{n} \tilde{l}_j x_j + \sum_{i=1}^{n}\sum_{j=1}^{n} \tilde{w}_{ij} y_{ij} && && (34)\\
\text{s.t.} \quad & \sum_{j=1}^{n} a_j x_j \leq b && && (35)\\
& \sum_{j=1}^{n} x_j = k && && (36)\\
& x_j \in \{0,1\} && j = 1,\ldots,n && (37)\\
& y_{ij} = x_i x_j && i,j = 1,\ldots,n && (38)\\
& x_j = \sum_{p\in\mathcal{P}} x_j^p \lambda^p && j = 1,\ldots,n && (39)\\
& y_{ij} = \sum_{p\in\mathcal{P}} y_{ij}^p \lambda^p && i,j = 1,\ldots,n && (40)\\
& \sum_{p\in\mathcal{P}} \lambda^p = 1 && && (41)\\
& \lambda^p \geq 0 && p \in \mathcal{P} &&
\end{aligned}
$$

Reformulation of ($kQKP^{conv}$) II

Constraints (47), (48) and (49) impose x and y to belong to a given polyhedron Ω, whose set of extreme points is denoted by $\mathcal{P}$. In our case, the following choices of Ω are possible:

$$\Omega_{\text{knap}} = \text{conv. hull}\{(x,y) : \sum_{j=1}^{n} a_j x_j \leq b, y_{ij} = x_i x_j, i,j = 1,\ldots,n, x_j \in \{0,1\}, j = 1,\ldots,n\}$$

$$\Omega_{\text{card}} = \text{conv. hull}\{(x,y) : \sum_{j=1}^{n} x_j = k, y_{ij} = x_i x_j, i,j = 1,\ldots,n, x_j \in \{0,1\}, j = 1,\ldots,n\}$$

$$\Omega_{\text{knap, card}} = \text{conv. hull}\{(x,y) : \sum_{j=1}^{n} a_j x_j \leq b, \sum_{j=1}^{n} x_j = k, y_{ij} = x_i x_j, i,j = 1,\ldots,n, x_j \in \{0,1\}, j = 1,\ldots,n\}$$

with $\mathcal{P}_{\text{knap}}$, $\mathcal{P}_{\text{card}}$, $\mathcal{P}_{\text{knap, card}}$ being the corresponding sets of extreme points.

Other reformulation of $(kQKP^{\text{conv}})$ after variables substitutions

$$
\begin{aligned}
(kQKP^{\text{conv}}_{\text{DWD}(\Omega)})\ \max \quad & \sum_{p\in\mathcal{P}}(\sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{q}_{ij}x_i^p x_j^p)\lambda^p + \sum_{p\in\mathcal{P}}(\sum_{j=1}^{n}\tilde{l}_j x_j^p)\lambda^p + \sum_{p\in\mathcal{P}}(\sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{w}_{ij} y_{ij}^p)\lambda^p && (42)\\
\text{s.t.} \quad & \sum_{p\in\mathcal{P}}(\sum_{j=1}^{n} a_j x_j^p)\lambda^p \le b && (43)\\
& \sum_{p\in\mathcal{P}}(\sum_{j=1}^{n} x_j^p)\lambda^p = k && (44)\\
& x_j \in \{0,1\} \qquad j=1,\dots,n && (45)\\
& y_{ij} \le x_i,\, y_{ij} \le x_j,\, y_{ij} \ge x_i + x_j - 1,\, y_{ij} \ge 0 \qquad i,j=1,\dots,n && (46)\\
& x_j = \sum_{p\in\mathcal{P}} x_j^p \lambda^p \qquad j=1,\dots,n && (47)\\
& y_{ij} = \sum_{p\in\mathcal{P}} y_{ij}^p \lambda^p \qquad i,j=1,\dots,n && (48)\\
& \sum_{p\in\mathcal{P}} \lambda^p = 1 && (49)\\
& \lambda^p \ge 0 \qquad p\in\mathcal{P}
\end{aligned}
$$

Formulations Overview

			Convexification		
			orig	conv	impr conv
	quad master	knap	NCM	$kQKP^{\text{conv}}_{\text{DWD}(knap)}$	$kQKP^{\text{impr. conv}}_{\text{DWD}(knap)}$
		card	NCM	IPP	IPP
Strenthening		knap+card	NCM	$kQKP^{\text{conv}}_{\text{DWD}(card,knap)}$	$kQKP^{\text{impr. conv}}_{\text{DWD}(card,knap)}$
	lin master	knap	$kQKP_{\overline{\text{DWD}}(knap)}$	$kQKP^{\text{conv}}_{\overline{\text{DWD}}(knap)}$	$kQKP^{\text{impr. conv}}_{\overline{\text{DWD}}(knap)}$
		card	$kQKP_{\overline{\text{DWD}}(card)}$	$kQKP^{\text{conv}}_{\overline{\text{DWD}}(card)}$	$kQKP^{\text{impr. conv}}_{\overline{\text{DWD}}(card)}$
		knap+card	POP	POP	POP

Hierarchy of reformulations

$$
\begin{array}{ccccc}
 & & kQKP^{\text{conv}}_{DWD(\cdot)} & \Leftarrow & kQKP^{\text{imp conv}}_{DWD(\cdot)} \\
 & & \Uparrow & & \Uparrow \\
kQKP_{\overline{DWD}(\cdot)} & \Leftarrow & kQKP^{\text{conv}}_{\overline{DWD}(\cdot)} & \Leftrightarrow & kQKP^{\text{imp conv}}_{\overline{DWD}(\cdot)}
\end{array}
$$

Experimental environment

- Carried out on an Intel i7-2600 quad core 3.4 GHz with 8 GB of RAM, using only one core
- CSDP integrated into COIN-OR for solving SDP programs
- CPLEX 12.6.2 with default settings
- Average values over 10 instances
- $n \in \{50, 60, \ldots, 100\}$
- $k \in [1, n/4]$, $b \in [50, 30k]$, a_j, $c_{ij} \in [1, 100]$

Numerical Results

n	$\delta(\%)$	$(kQKP^{conv})$ Gap	Time	$(kQKP^{impr\ conv})$ Gap	Time	$(kQKP^{conv}_{DWD(knap)})$ Gap	Time	$(kQKP^{impr\ conv}_{DWD(knap,\ card)})$ Gap	Time
50	*25*	102.65	0.02	30.89	1.05	38.36	1.97	29.15	9.30
	50	150.56	0.06	25.25	0.94	31.05	3.04	23.66	9.71
	75	230.29	0.12	105.16	1.09	114.26	1.55	100.88	8.06
60	*25*	60.76	0.04	130.92	0.04	149.25	1.55	126.07	10.89
	50	93.73	0.11	15.08	2.61	19.48	3.86	14.19	19.05
	75	212.67	0.25	141.08	2.09	151.22	1.67	136.22	8.99
70	*25*	130.23	0.06	38.03	5.11	46.84	4.82	36.52	33.25
	50	177.07	0.19	72.81	4.27	80.44	6.83	70.77	54.37
	75	382.36	0.44	56.26	3.45	63.77	3.25	54.57	22.19
80	*25*	111.24	0.08	34.05	7.98	41.90	5.64	32.87	71.19
	50	271.64	0.26	55.44	9.59	64.09	4.67	53.65	43.98
	75	313.33	0.66	83.58	7.42	92.31	4.64	81.47	43.42
90	*25*	118.45	0.13	112.80	13.75	129.31	4.74	109.63	44.66
	50	248.57	0.48	83.15	12.38	92.19	4.52	81.65	66.75
	75	388.68	1.06	37.90	5.63	42.13	6.95	37.12	102.54
100	*25*	169.43	0.16	73.90	23.49	82.78	6.80	72.72	99.90
	50	145.72	0.49	17.38	28.06	21.83	8.58	17.19	219.77
	75	260.26	1.25	21.67	18.37	27.22	6.23	21.50	158.30
Avg	*25*	115.46	0.08	70.10	8.57	81.41	4.25	67.83	44.86
	50	181.22	0.26	44.85	9.64	51.51	5.25	43.52	68.94
	75	297.93	0.63	74.28	6.34	81.82	4.05	71.96	57.25
Avg		198.20	0.32	63.08	8.18	71.58	4.52	61.10	57.02

Numerical Results

		$(kQKP^{\text{conv}})$		$(kQKP^{\text{impr conv}}_{\text{DWD(knap, card)}})$		$kQKP^{\text{conv}}_{\overline{\text{DWD}}(knap)}$		$kQKP^{\text{conv}}_{\overline{\text{DWD}}(knap)}$	
n	$\delta(\%)$	Gap	Time	Gap	Time	Gap	Time	Gap	Time
50	*25*	102.65	0.02	29.15	9.30	19.43	0.30	0.00	40.13
	50	150.56	0.06	23.66	9.71	28.97	0.08	0.00	34.72
	75	230.29	0.12	100.88	8.06	119.44	0.05	0.00	14.65
60	*25*	60.76	0.04	126.07	10.89	79.77	0.14	0.00	27.93
	50	93.73	0.11	14.19	19.05	13.13	0.25	0.00	32.54
	75	212.67	0.25	136.22	8.99	157.01	0.07	0.00	16.89
70	*25*	130.23	0.06	36.52	33.25	16.80	1.20	0.00	64.44
	50	177.07	0.19	70.77	54.37	47.85	4.30	0.00	72.07
	75	382.36	0.44	54.57	22.19	65.16	0.15	0.00	55.08
80	*25*	111.24	0.08	32.87	71.19	11.67	210.59	0.00	78.04
	50	271.64	0.26	53.65	43.98	42.69	0.77	0.00	66.70
	75	313.33	0.66	81.47	43.42	95.92	14.00	0.00	85.52
90	*25*	118.45	0.13	109.63	44.66	57.50	1692.40	0.00	89.42
	50	248.57	0.48	81.65	66.75	63.79	190.64	0.00	102.25
	75	388.68	1.06	37.12	102.54	59.35	15.16	0.00	491.36
100	*25*	169.43	0.16	72.72	99.90	41.07	926.10	0.00	145.81
	50	145.72	0.49	17.19	219.77	14.75	577.86	0.00	289.12
	75	260.26	1.25	21.50	158.30	47.82	3.77	0.00	656.70
Avg	*25*	115.46	0.08	67.83	44.86	37.71	471.79	0.00	74.30
	50	181.22	0.26	43.52	68.94	35.20	128.98	0.00	99.57
	75	297.93	0.63	71.96	57.25	90.79	5.53	0.00	220.03
Avg		198.20	0.32	61.10	57.02	54.56	202.10	0.00	131.30

The best reformulation I

$$
\begin{aligned}
kQKP^{\text{conv}}_{\overline{\text{DWD}}(knap)} \quad \max \quad & \sum_{p \in \mathcal{P}_{\text{knap}}} c^p_{u^*,v^*} \lambda^p \\
\text{s.t.} \quad & x_j = \sum_{p \in \mathcal{P}_{\text{knap}}} x^p_j \lambda^p && j = 1, \ldots, n && [\phi_j] \\
& \sum_{p \in \mathcal{P}_{\text{knap}}} \sum_{j=1}^{n} x^p_j \lambda^p = k && && [\gamma] \\
& \sum_{p \in \mathcal{P}_{\text{knap}}} \lambda^p = 1 && && [\theta] \\
& \lambda^p \geq 0 && p \in \mathcal{P}_{\text{knap}}
\end{aligned}
$$

with γ and θ the dual variables.

The best reformulation II

Denoting as γ^* and θ^* the optimal dual variables of a restricted master solution during a column generation iteration, the pricing problem can be written as follows:

$$\begin{aligned}
\max \quad & f_{u^*,v^*}(x) + \sum_{i=1}^{n} \phi_i^* x_i + \gamma^* + \theta^* \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_j x_j \leq b \\
& x_j \in \{0,1\} \qquad\qquad j = 1, \ldots, n
\end{aligned}$$

Outline

Settings

The problem

$$\begin{aligned} \min \quad & f(x) = x^T Q x + c^T x \\ s.\ t. \quad & x \in X. \end{aligned}$$

where $x \in \mathbb{R}^n$, $X \subset \mathbb{R}^n$.

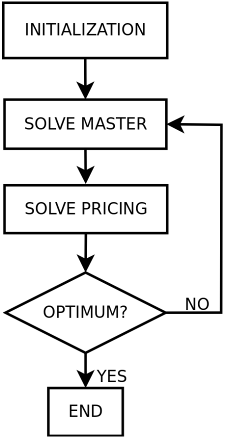
Hypotheses:

- $Q \in \mathbb{R}^{n \times n}$ is positive semidefinite and dense.
- X is a polytope.
- High number of variables and low number of constraints.

A column generation method

Simplicial Decomposition (SD)

- Master problem: original objective function, optimized over a simplex.
- Pricing problem: linear objective function, original domain.
- All the original constraints are in the pricing.
- Finite convergence.



The master problem

At a k-th iteration, k vertices $x_1, \dots x_k \in X$ are provided. $k \ll n$.

Master problem

$$\begin{aligned} \min \quad & x^T Q x + c^T x \\ \text{s. t.} \quad & x = \sum_{i=1}^{k} \omega_i x_i, \\ & \sum_{i=1}^{k} \omega_i = 1, \\ & \omega_i \geq 0, \quad \forall i = 1, \dots, k. \end{aligned}$$

The pricing problem

Pricing problem

$$\begin{aligned} \min \quad & \nabla f(x_m)^T x \\ \text{s. t.} \quad & x \in X. \end{aligned}$$

- Linearization of the original objective function in the optimal point x_m of the master.
- Same dimension as the original problem.
- Same constraints as the original problem.

Master: SD - ACDM

Adapted Conjugate Direction Method (ACDM)

Main ideas:

- Based on the conjugate Directions Method.
- Reuse the informations from previous iteration.
- Exploit the special structure of the simplices generated.

The Conjugate Direction Method

- Two directions $d_1, d_2 \in \mathbb{R}^k$ are conjugated with respect to the positive definite quadratic matrix $Q \in \mathbb{R}^{k \times k}$ if: $d_1^T Q d_2 = 0$.
- If we have a set of k conjugate directions $D = \{d_1, \ldots, d_k\}$, the minimum of $f(x) = x^T Qx + c^T x$ can be found in k steps by optimizing in sequence over the k conjugate directions.

Master: SD - ACDM

Adapted Conjugate Direction Method (ACDM)

- Reuse the information from the previous conjugate directions.
- Exploit the special structure of the simplices generated.

Main Steps

- At iteration k, we have a set of $k-1$ conjugate directions D from the previous iteration.
- The pricing provides a new point x_k (i.e., a new dimension).
- Find a new direction d_k connecting $x_{k-1}$with x_k and conjugate it w.r.t. the set D.
- Find new optimal point along this direction.

PRO: Most of the times, only one step.
CON: If the optimum is on a face, all the directions must be recalculated.

Master: SD - FGPM

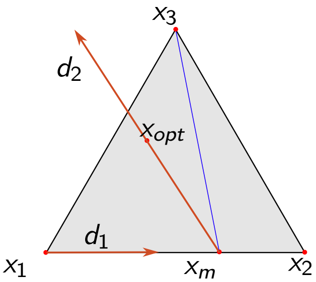
A Fast Gradient Projection Method, (FGPM)

A more general method based on the projected gradient approach.
Warmstart: start in the previous optimal point.
Iteratively, given the k-th point $\tilde{x_k}$:

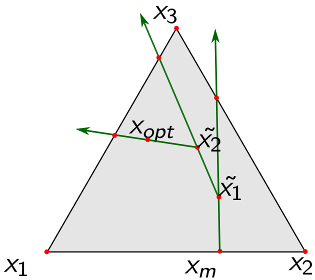
- compute the gradient $\nabla f(\tilde{x_k})$;
- project the point $y_k = \tilde{x_k} - s\nabla f(\tilde{x_k})$ onto the simplex;
- if $y_k \neq \tilde{x_k}$, find α_k with an Armijo-like rule;
- compute $\tilde{x_{k+1}} = \tilde{x_k} + \alpha_k(p(y_k) - \tilde{x_k})$.

Master solvers

ACDM



FGPM



Pricing improvements I

Adding cuts

- Reduce the search region;
- exclude vectors that give ascent directions with respect to the previous partial optima;
- add cuts of the form

$$\nabla f(x_m^i)^T(x - x_m^i) \leq 0, \quad \exists i \in \{1, \dots, k-1\}.$$

Early stopping

- Stop the computation before reaching the optimum, but ensure a descent direction: generate the point $\bar{x}_k$ s. t.

$$\nabla f(x_k)^T(\bar{x}_k - x_k) \leq -\varepsilon < 0.$$

Pricing improvements II

Sifting

Consider the *Sifting* options for the Cplex solver in addition to the default primal simplex.

Sifting is a column generation algorithm:

- it solves the problem with a (small) subset of columns;
- it evaluates the reduced costs of the remaining columns;
- columns that violate the optimality condition are inserted.

Problem instances

Portfolio optimization problem
(Markowitz's formulation)
(Literature data)

$$\begin{aligned} \min \;& f(x) = x^T \Sigma x \\ \text{s. t.} \;& r^T x \geq \mu, \\ & e^T x = 1, \\ & x \geq 0. \end{aligned}$$

General quadratic problems
(Randomly generated)

$$\begin{aligned} \min \;& f(x) = x^T Q x + c^T x \\ \text{s. t.} \;& Ax \geq b, \\ & 0 \leq x \leq 1. \end{aligned}$$

Problem instances

Quadratic shortest path problems
(Literature and randomly generated data)

$$\begin{aligned} \min\ & f(x) = x^\top Q x + c^\top x \\ \text{s. t.}\ & \sum_{e \in \delta^+(s)} x_s = 1, \\ & \sum_{e \in \delta^+(v)} x_v - \sum_{e \in \delta^-(v)} x_v = 0, \quad \forall v \neq s, t \\ & \sum_{e \in \delta^-(t)} x_t = 1. \end{aligned}$$

Multidimensional quadratic knapsack problem
(Literature and randomly generated data)

$$\begin{aligned} \min\ & f(x) = x^\top Q x + c^\top x \\ \text{s. t.}\ & \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad \forall i = 1, 2, \ldots, m \\ & 0 \leq x \leq 1. \end{aligned}$$

Instances

- Portfolio Optimization (PO): 40 instances, dimension 225 to 10980.
- General quadratic :
  - small m (GS): 450 instances:
    $n = 2000$ to 10000, $m = 2$ to 42.
  - large m (GL): 750 instances:
    $n = 2000$ to 10000, $m = n/32$ to $n/2$.
- Quadratic shortest path problems : grid and random shortest path instances ($1000 \leq n \leq 10000$) : 102 instances.
- Multidimensional quadratic knapsack problem : 54 instances:
  - ORLib dataset and GK dataset ($n \geq 1000$).
  - Randomly generated instances ($5000 \leq n \leq 10000$).

Hardware and software

1. *IBM Ilog Cplex v.12.6.2.*

Results: Portfolio optimization problem I

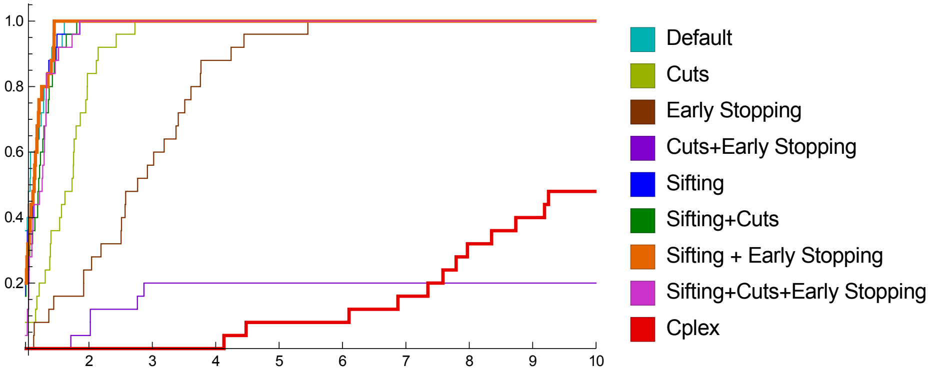


Figure: Performance profile PO, pricing options

Results: Portfolio optimization problem II

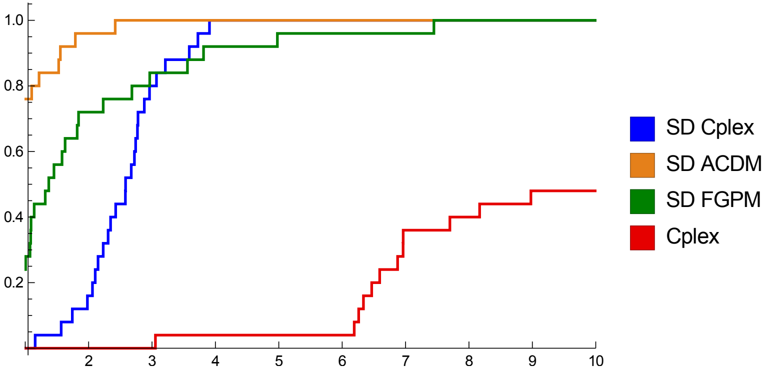


Figure: Performance profile PO, master solvers

Results: General quadratic problems I

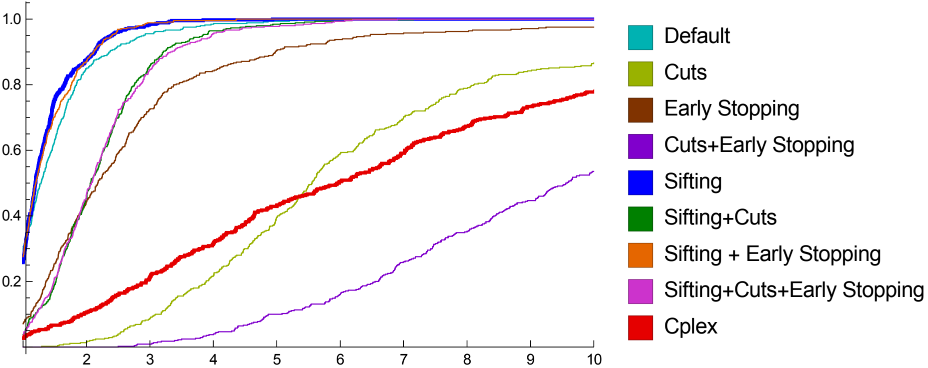


Figure: Performance profile GS, pricing options

Results: General quadratic problems II

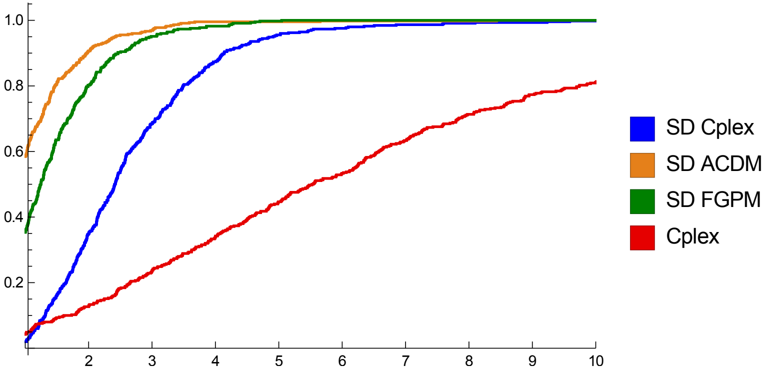


Figure: Performance profile GS, master solvers

Results: General quadratic problems III

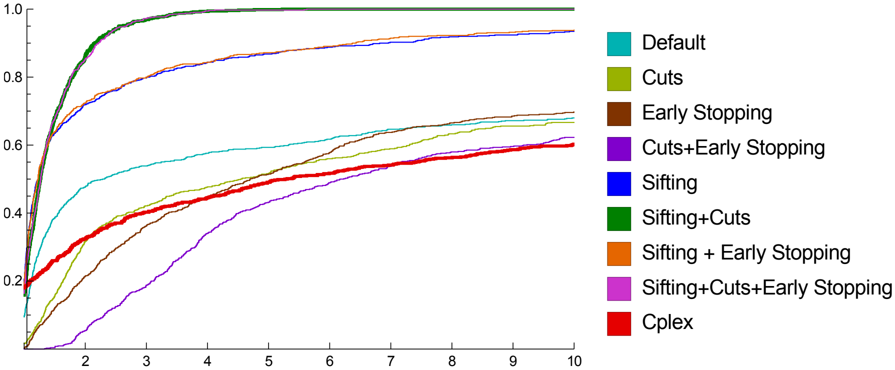


Figure: Performance profile GL, pricing options

Results: General quadratic problems IV

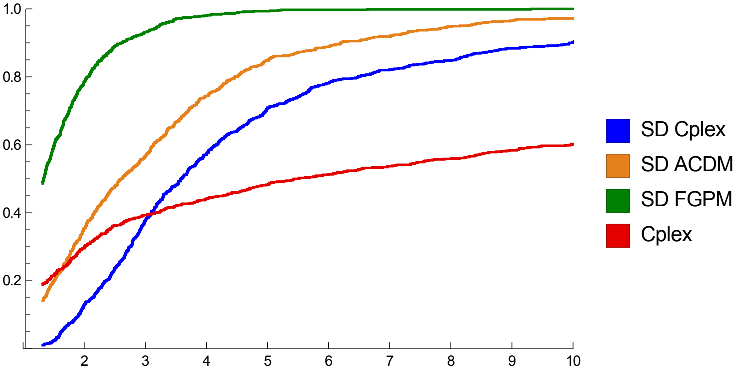


Figure: Performance profile GL, master solvers

Results: Quadratic shortest path problem I

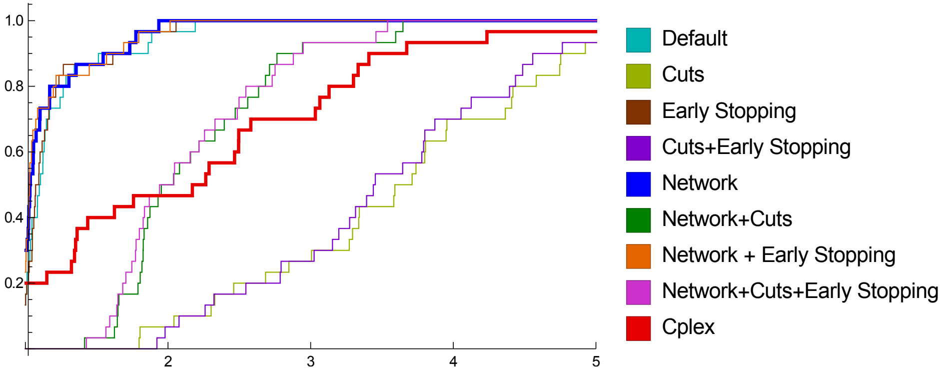


Figure: Performance profiles for grid shortest path instances - pricing options (SD FGPM).

Results: Quadratic shortest path problem II

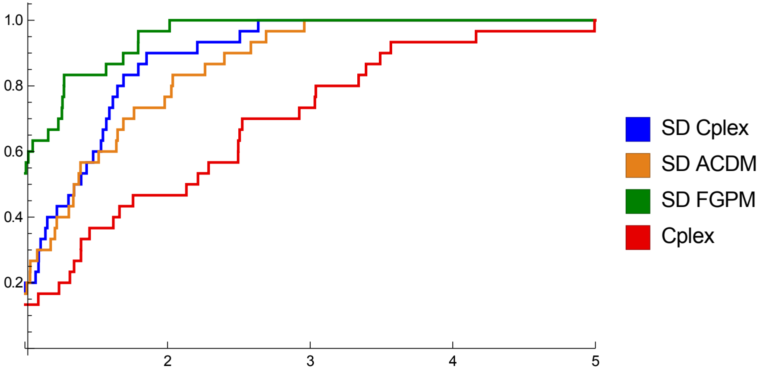


Figure: Performance profiles for grid shortest path instances - master solvers.

Results: Quadratic shortest path problem III

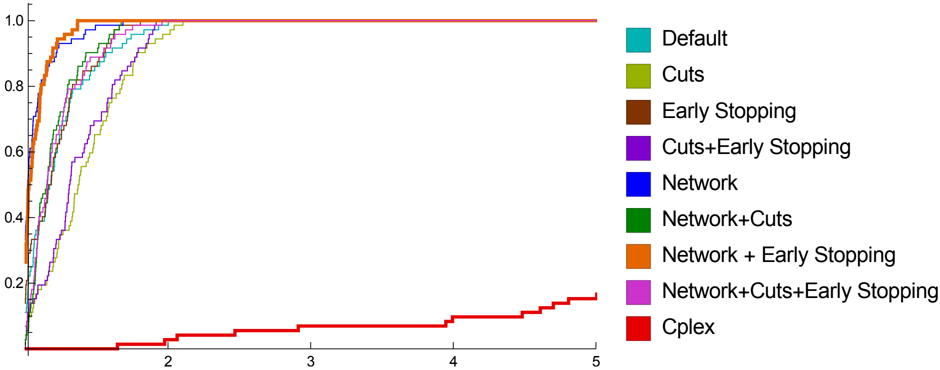


Figure: Performance profile for random shortest path instances - pricing options (SD ACDM).

Results: Quadratic shortest path problem IV

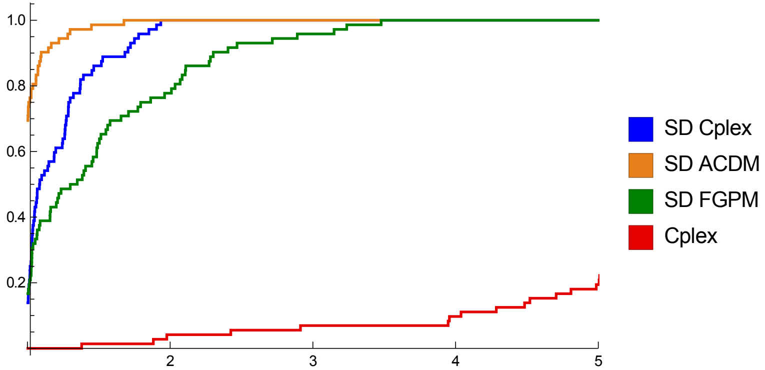


Figure: Performance profile for random shortest path instances - master solvers.

Results: Multidimensional quadratic knapsack problem I

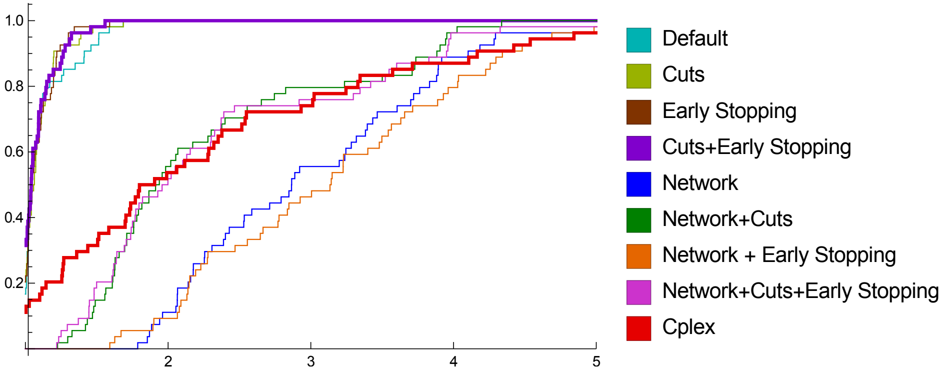


Figure: Performance profile for multidimensional knapsack instances - pricing options (SD ACDM).

Results: Multidimensional quadratic knapsack problem II

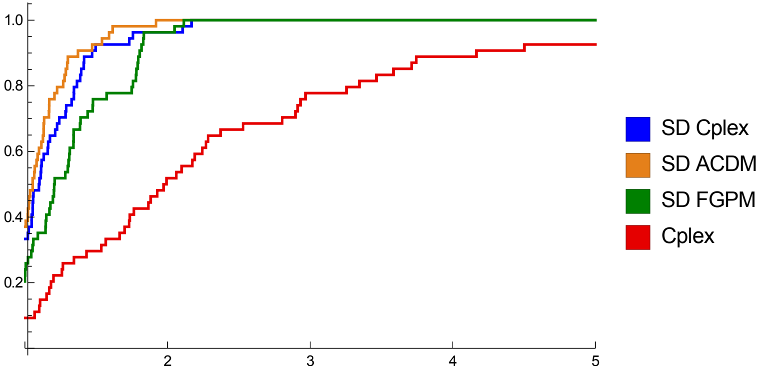


Figure: Performance profile for multidimensional knapsack instances - master solvers.

The model

Generic formulation

$$\min \ f(x) = x^\top \bar{Q} x + q^\top x$$
$$\text{s. t.} \quad x^\top \bar{A}_i x + a_i^\top x \leq b_i, \quad \forall i = 1 \ldots, m$$
$$x \in \{0,1\}^n.$$

$$x \in \mathbb{R}^n, n \in \mathbb{N},$$
$$\bar{Q},\ \bar{A}_i \in \mathcal{S}^n$$
$$q,\ a_i \ \in \mathbb{R}^n$$
$$b_i \in \mathbb{R},$$

The model

Generic formulation

$$\min \; f(x) = x^\top \bar{Q} x + q^\top x$$
$$\text{s. t.} \quad x^\top \bar{A}_i x + a_i^\top x \le b_i, \quad \forall i = 1 \dots, m$$
$$x \in \{0,1\}^n.$$

$x \in \mathbb{R}^n, n \in \mathbb{N},$
$\bar{Q},\ \bar{A}_i \in \mathcal{S}^n$
$q,\ a_i \ \in \mathbb{R}^n$
$b_i \in \mathbb{R},$

Compact formulation

Since $x_i^2 = x_i$:
Let $Q = \bar{Q} + \mathrm{I}\, q,\ A_i = \bar{A}_i + \mathrm{I}\, a_i \ \rightarrow$

$$\min x^\top Q x$$
$$\text{s. t.} \quad x^\top A_i x \le b_i, \quad \forall i$$
$$x \in \{0,1\}^n.$$

Matrix space

The problem can be written in matrix form:

$$\langle M, X\rangle := \mathrm{Tr}(M^\top X).$$

Extended formulation

$$\begin{aligned} &\min \langle Q, X\rangle \\ \text{s. t.} \quad &\langle A_i, X\rangle \leq b_i, \quad \forall i = 1\ldots, m \\ &X = xx^T \\ &x \in \{0,1\}^n \end{aligned}$$

Relaxing constraint

We relax the constraint

$$X = xx^T$$

and let X be in the *convex hull* of 0-1 rk-1 matrices:

$$X = \sum_{p=1}^{2^n} x_p x_p^\top \lambda_p$$

$$\sum_{p=1}^{2^n} \lambda_p = 1$$

$$\lambda \geq 0$$

$$x_p \in \{0,1\}^n.$$

Relaxing constraint

We relax the constraint

$$X = xx^T$$

and let X be in the *convex hull* of 0-1 rk-1 matrices:

$$X = \sum_{p=1}^{2^n} x_p x_p^\top \lambda_p$$

$$\sum_{p=1}^{2^n} \lambda_p = 1$$

$$\lambda \geq 0$$

$$x_p \in \{0,1\}^n.$$

Definition: (Restricted) Boolean Quadric Polytope (BQP) of size n

$$BQP_n = Conv\left\{X \in \mathbb{R}^{n\times n} \,|\, X = xx^\top,\ x \in \{0,1\}^n\right\}$$

Some relations

The CP and PSD cones:

We recall that the Completely Positive (CP) and the Positive Semi Definite (PSD) cones are respectively:

$$CP_n = Conv\left\{X \in \mathbb{R}^{n\times n} \,|\, X = xx^\top,\ x \in \mathbb{R}^n,\ x \geq 0\right\}$$
$$PSD_n = Conv\left\{X \in \mathbb{R}^{n\times n} \,|\, X = xx^\top,\ x \in \mathbb{R}^n\right\}$$

Then,

$$BQP_n \subset CP_n \subset PSD_n.$$

Lower bounds

Hence the lower bound (LB) obtained with our relaxation is stronger than the CP and PSD bounds:

$$LB_{BQP} \geq LB_{CP} \geq LB_{PSD}.$$

A column generation algorithm

Let $\mathcal{P} := \{1, \ldots, 2^n\}$. Then, we have:

Formulation

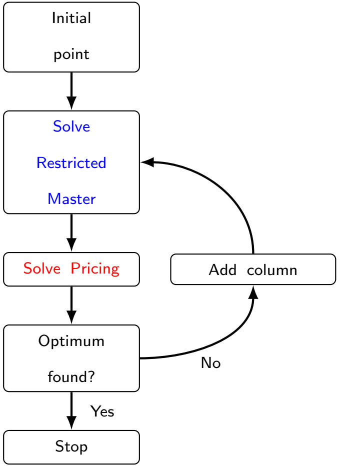
$$
(1) \quad \begin{aligned}
\min\ & \langle Q, X \rangle \\
\text{s. t. } & \langle A_i, X \rangle \leq b_i, \quad \forall i = 1 \ldots, m \\
& X = \sum_{p \in \mathcal{P}} \bar{X}_p \lambda_p \\
& \sum_{p \in \mathcal{P}} \lambda_p = 1 \\
& \lambda_p \geq 0 \quad \forall p \in \mathcal{P} \\
& \bar{X}_p = \bar{x}_p \bar{x}_p^\top \quad \forall p \in \mathcal{P} \\
& \bar{x}_p \in \{0, 1\}^n \quad \forall p \in \mathcal{P}.
\end{aligned}
$$

A column generation algorithm

Let $\mathcal{P} := \{1, \ldots, 2^n\}$. Then, we have:

Formulation

$$
(1) \quad \begin{aligned}
\min \ & \langle Q, X \rangle \\
\text{s. t.} \ & \langle A_i, X \rangle \leq b_i, \quad \forall i = 1 \ldots, m \\
& X = \sum_{p \in \mathcal{P}} \bar{X}_p \lambda_p \\
& \sum_{p \in \mathcal{P}} \lambda_p = 1 \\
& \lambda_p \geq 0 \quad \forall p \in \mathcal{P} \\
& \bar{X}_p = \bar{x}_p \bar{x}_p^\top \quad \forall p \in \mathcal{P} \\
& \bar{x}_p \in \{0, 1\}^n \quad \forall p \in \mathcal{P}.
\end{aligned}
$$

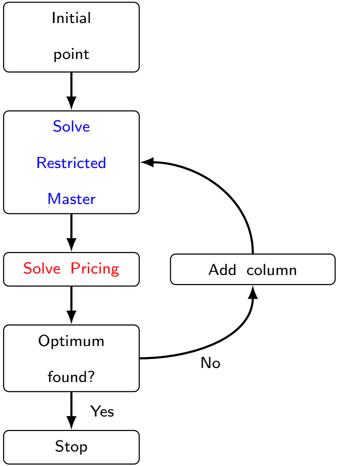


A column generation algorithm

Let $\mathcal{P} := \{1, \dots, 2^n\}$. Then, we have:

Formulation

$$
(1) \quad \begin{aligned}
\min \ & \langle Q, X \rangle \\
\text{s. t. } & \langle A_i, X \rangle \leq b_i, \quad \forall i = 1 \dots, m \\
& X = \sum_{p \in \mathcal{P}} \bar{X}_p \lambda_p \\
& \sum_{p \in \mathcal{P}} \lambda_p = 1 \\
& \lambda_p \geq 0 \quad \forall p \in \mathcal{P} \\
& \bar{X}_p = \bar{x}_p \bar{x}_p^\top \quad \forall p \in \mathcal{P} \\
& \bar{x}_p \in \{0, 1\}^n \quad \forall p \in \mathcal{P}.
\end{aligned}
$$



Master and Pricing problems

Let $\bar{\mathcal{P}} \subset \mathcal{P}$, $\quad \bar{X}_p := x_p x_p^\top,\ p \in \bar{\mathcal{P}}$

Restricted Master Problem (RMP)

$$\begin{aligned}
\min\ & \langle Q, X \rangle \\
\text{s. t.}\ \ & \langle A_i, X \rangle \le b_i, \quad \forall i = 1 \ldots, m \\
& X = \sum_{p \in \bar{\mathcal{P}}} \bar{X}_p \lambda_p \\
& \sum_{p \in \bar{\mathcal{P}}} \lambda_p = 1 \\
& \lambda_p \ge 0 \quad \forall p \in \bar{\mathcal{P}}.
\end{aligned}$$

Master and Pricing problems

RMP,reduced form

$$\begin{aligned}
\min \quad & \sum_{p \in \bar{\mathcal{P}}} \langle Q, X_p \rangle \lambda_p \\
\text{s. t.} \quad & \sum_{p \in \bar{\mathcal{P}}} \langle A_i, X_p \rangle \lambda_p \leq b_i, \quad \forall i = 1 \ldots, m \qquad [\pi] \\
& \sum_{p \in \bar{\mathcal{P}}} \lambda_p = 1 \qquad [\pi_0] \\
& \lambda_p \geq 0 \quad \forall p \in \bar{\mathcal{P}}.
\end{aligned}$$

Master and Pricing problems

Dual problem

$$\begin{aligned} \max\ & b^\top \pi + \pi_0 \\ \text{s. t.}\ & \sum_{i=1}^{m} \langle A_i, \bar{X}_p \rangle \pi_i + \pi_0 \leq \langle Q, \bar{X}_p \rangle, \quad \forall p \in \bar{\mathcal{P}} \\ & \pi \leq 0 \end{aligned}$$

Pricing problem

$$\begin{aligned} \min\ & \langle Q, X \rangle - \sum_{i=1}^{m} \langle A_i, X \rangle \pi_i^* - \pi_0^* \\ \text{s. t.}\ & X = xx^\top \\ & x \in \{0,1\}^n \end{aligned}$$

CP reformulation for Binary Quadratic Problems

The following problems are equivalent (Burer, 2009):

Binary Quadratic Problems

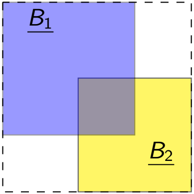
$$\begin{aligned} \min \quad & x^\top Q x + q^\top x \\ \text{s. t.} \quad & a_i^\top x = b_i, \quad \forall i = 1\ldots, m \\ & x \in \{0,1\}^n. \end{aligned}$$

Completely Positive reformulation

$$\begin{aligned} \min \ & \langle Q, \bar{X} \rangle + q_0^\top x \\ \text{s. t.} \quad & a_i^\top x = b_i, \quad \forall i = 1\ldots, m \\ & a_i^\top \bar{X} a_i = b_i^2, \quad \forall i = 1\ldots, m \\ & x_j = \bar{X}_{jj} \quad \forall j = 1, \ldots n \\ & \begin{pmatrix} 1 & x^\top \\ x & \bar{X} \end{pmatrix} \in CP_{n+1}. \end{aligned}$$

Hence, we have an exact reformulation for binary QPs with linear equalities (no branching needed).

Notations



Let:

- $\underline{b}_j \subset \{1, \ldots, n\} \, \forall j = 1, \ldots, k$;
- $\bigcup_{j=1}^{k} \underline{b}_j = \{1, \ldots, n\}$;
- $\underline{B}_j = \underline{b}_j \times \underline{b}_j \, \forall j = 1, \ldots, k$.

A Block structure is $\underline{\mathcal{B}_k} = \bigcup_{j=1}^{k} \underline{B}_j$.

Block decomposable problems

A problem is block decomposable if all nonzero entries of Q, A_i belong to $\underline{\mathcal{B}_k}$.

$\forall X \in \mathbb{R}^{n \times n}$ we indicate with

$$X^{B_j} := \{X_{p,q} | p, q \in \underline{b}_j\} \in \mathbb{R}^{d_j \times d_j} \quad \forall j = 1, \ldots, k$$

the restriction of X to a block j; d_j is the dimension of $\underline{b}_j$.
$\forall j = 1, \ldots k$, for $M = Q, A_i \; \forall i = 1, \ldots m$ let $M^j := M^{B_j}$, with
$M_{p,q} = 0 \; \forall p, q \in \underline{b}_j \setminus (\underline{b}_1, \ldots, \underline{b}_{j-1})$.

Block decomposition

Block-decomposed Master Program formulation

$$
(2) \quad \begin{aligned}
\min \quad & \sum_{j=1}^{k} \langle Q^j, Y_j \rangle \\
\text{s. t.} \quad & \sum_{j=1}^{k} \langle A_i^j, Y_j \rangle \leq b_i, \quad \forall i = 1 \ldots, m \\
& Y_j^{B_j \cap B_h} = Y_h^{B_j \cap B_h} \quad \forall 1 \leq j < h \leq k \\
& Y_j = \sum_{l=1}^{2^{d_j}} \mu_l^j (y_j^l)(y_j^l)^\top \quad \forall j = 1, \ldots, k \\
& \sum_{l=1}^{2^{d_i}} \mu_l^j = 1 \quad \forall j = 1, \ldots, k \\
& \mu_l^j \geq 0 \quad \forall l = 1, \ldots 2^{d_j}, \forall j = 1, \ldots, k. \\
& y_j^l \in \{0,1\}^{d_j} \quad \forall l = 1, \ldots, 2^{d_j} \, \forall j = 1, \ldots, k.
\end{aligned}
$$

Block-decomposed restricted master and pricing

Let $\bar{\mathcal{P}}_j \subseteq \{1, \dots, 2^{d_j}\}\ \forall j = 1, \dots, k$. Then:

Block-decomposed RMP

$$
\begin{aligned}
\min\ & \sum_{j=1}^{k} \langle Q^j, Y_j \rangle \\
\text{s. t.}\ & \sum_{j=1}^{k} \langle A_i^j, Y_j \rangle \le b_i, \quad \forall i = 1 \dots, m \qquad [\alpha] \\
& Y_j^{B_j \cap B_h} = Y_h^{B_j \cap B_h} \quad \forall 1 \le j < h \le k \quad [\beta^{j,h}] \\
& Y_j = \sum_{l \in \bar{\mathcal{P}}_j} \mu_l^j (y_j^l)(y_j^l)^\top \quad \forall j = 1, \dots, k \quad [\pi^j] \\
& \sum_{l \in \bar{\mathcal{P}}_j} \mu_l^j = 1 \quad \forall j = 1, \dots, k \qquad [\pi_0^j] \\
& \mu_l^j \ge 0 \quad \forall l \in \bar{\mathcal{P}}_j, \forall j = 1, \dots, k.
\end{aligned}
$$

Block-decomposed restricted master and pricing

Dual problem

$$\max \; b^\top \alpha + \sum_{j=1}^{k} \pi_0^j$$

$$\text{s. t. } \sum_{i=1}^{m} A_i^j \alpha_i + \sum_{h=1, h>j}^{k} C^{j,h} \beta^{j,h} - \sum_{h=1, h<j}^{k} C^{j,h} \beta^{j,h} + \pi^j = Q^j \quad \forall j = 1, \ldots, k$$

$$- \langle (y_j^l)(y_j^l)^\top, \pi^j \rangle + \pi_0^j \leq 0, \quad \forall l \in \bar{\mathcal{P}}$$

$$\alpha \leq 0,$$

where $(C^{j,h})_{p,q} = 1$ if $(p,q) \in \underline{B}_j \cap \underline{B}_h$, 0 otherwise.

Pricing problems

$$\min \; \langle {\pi^j}^*, Y_j \rangle - {\pi_0^j}^*$$

$$\text{s. t. } Y_j = y_j y_j^\top$$

$$y_j \in \{0,1\}^{d_j} \quad \forall j = 1, \ldots, k.$$

Equivalence problem between (1) and (2)

First inclusion "$\supseteq$"

If X, λ are feasible for (1), $\exists$ Y_j, μ_j^l feasible for (2), s.t. $X^{B_j} = Y_j \, \forall j = 1, \ldots, k$?

Equivalence problem between (1) and (2)

First inclusion "$\supseteq$"

If X, λ are feasible for (1), $\exists$ Y_j, μ_j^l feasible for (2), s.t. $X^{B_j} = Y_j \, \forall j = 1, \ldots, k$?

Answer

Yes, always. Hence (2) always gives a valid lower bound.

Equivalence problem between (1) and (2)

First inclusion "$\supseteq$"

If X, λ are feasible for (1), $\exists$ Y_j, μ_j^l feasible for (2), s.t. $X^{B_j} = Y_j \, \forall j = 1, \dots, k$?

Answer

Yes, always. Hence (2) always gives a valid lower bound.

Second inclusion "$\subseteq$"

If Y_j, μ_j^l are feasible for (2), $\exists$ X, λ feasible for (1), s.t. $X^{B_j} = Y_j \, \forall j = 1, \dots, k$?

Equivalence problem between (1) and (2)

First inclusion "$\supseteq$"

If X, λ are feasible for (1), $\exists$ Y_j, μ_j^l feasible for (2), s.t. $X^{B_j} = Y_j \, \forall j = 1, \ldots, k$?

Answer

Yes, always. Hence (2) always gives a valid lower bound.

Second inclusion "$\subseteq$"

If Y_j, μ_j^l are feasible for (2), $\exists$ X, λ feasible for (1), s.t. $X^{B_j} = Y_j \, \forall j = 1, \ldots, k$?

Answer

It depends on the block structure.

Counterexample

Let Q, $A_i \in \mathbb{R}^{4\times 4}$ have the following block structure:

$$\underline{b_1} = \{1,2\}, \underline{b_2} = \{2,3\}, \underline{b_3} = \{3,4\}, \underline{b_4} = \{1,4\}.$$

Then:

A feasible solution for (2)

$$\frac{1}{2}\begin{pmatrix} 1 & 1 & & 0 \\ 1 & 1 & 1 & \\ & 1 & 1 & 1 \\ 0 & & 1 & 1 \end{pmatrix},$$

given by:

$$Y_j = \frac{1}{2}\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad j = 1,2,3$$

$$Y_4 = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix},$$

it cannot be *completed* to a solution to (1).

BQP completion problem

Definitions

Let $M \in \mathbb{R}^{n \times n}$, symmetric.

- M is partial if some entries are not specified;
- a specification graph of M has n vertices and edges $\{i, j\}$ if $M_{i,j}$ is *specified*;
- M is partial BQP if $\forall$ fully specified principal submatrix N, $N \in BQP$;
- if M is *partial BQP*, it is BQP completable if $\exists N \in BQP_n$ fully specified, $N_{i,j} = M_{i,j}$ where $M_{i,j}$ is specified;
- a graph G is BQP completable if *every* partial BQP matrix with specification graph G is BQP completable.

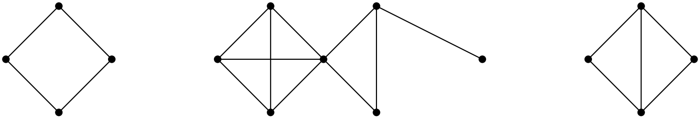
BQP completion problem $\leftrightarrow$ "$\subseteq$" inclusion:

Which graphs are BQP completable?

Theoretical results

Known results: PSD and CP completion problems

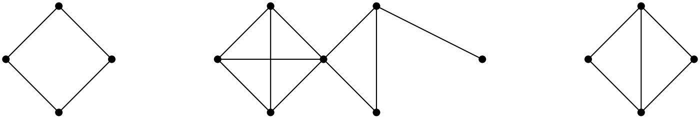
- A graph is PSD-completable iff it is chordal;
- A graph is CP-completable iff it is block-clique.



Theoretical results

Known results: PSD and CP completion problems

- A graph is PSD-completable iff it is chordal;
- A graph is CP-completable iff it is block-clique.



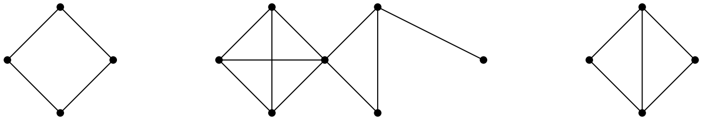
BQP completion problem

- If G is not chordal, it is not BQP-completable;
- If G is chordal, is it BQP-completable?

Theoretical results

Known results: PSD and CP completion problems

- A graph is PSD-completable iff it is chordal;
- A graph is CP-completable iff it is block-clique.



BQP completion problem

- If G is not chordal, it is not BQP-completable;
- If G is chordal, is it BQP-completable?
  - if the max size d of intersections is 2: yes;
  - if $d > 2$: work in progress.

Preliminary results

Instances

We selected some instances from the *QPlib* library.
We compared with the root node bound provided by SDP relaxation:
SDP solver: *BiqCrunch* (BC)

Results

Instance	Opt val	BC-bound		BC-cuts		CP-base		CP-blocks	
		Bound	T (s)	Bound	T (s)	Bound	T (s)	Bound	T (s)
QPLIB-1976	-9594	-51092	41	-45075	324	-44898	7	-44898	0.14
QPLIB-2017	-22984	-83215	490	-78525	1609	-78215	433	-78215	0.42
QPLIB-2029	-34704	-220262	856	-220262	900	-101334	2128	-101334	0.54
QPLIB-2036	-30590	-136227	1006	-127166	3287	-126386	391	-126386	0.07
QPLIB-2055	3389110	1999554	21	2209752	104	2314020	92	-	t.l.
QPLIB-2060	2528144	1466569	36	1703346	655	1707160	153	-	t.l.
QPLIB-2085	7034580	4705157	85	5420526	2642	5432400	1066	-	t.l.
QPLIB-2096	7068000	5826148	82	6305261	2679	6312620	1210	-	t.l.

Table: Root node bound and time for QCQP instances.