The p-Lagrangian method for MIQCQPs

Fabricio Oliveira

Department of Mathematics and Systems Analysis
School of Science, Aalto University

ISMP 2018 - Bordeaux, France

July 5, 2018

The team

Tiago Andrade

Fabricio Oliveira

Silvio Hamacher

Andrew Eberhard

Outline of this talk

Outline of this talk

Introduction

(R)NMDT and MILP relaxation

p-Lagrangian decomposition

Numerical results

Conclusions and future directions

Refinery Operations Planning Problem

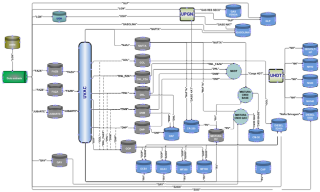
Oil refinery operational planning

- Seeks to maximize profit;
- Several possible configurations;
- Product property specifications must be met;

Model characteristics:

- Bilinear (nonconvex) and mixed-integer;
- Large number of flows;
- Several nonlinear constraints.

Lit. references: Neiro and Pinto [9, 10], and Andrade et al. [2].



Refinery Operations Planning Problem

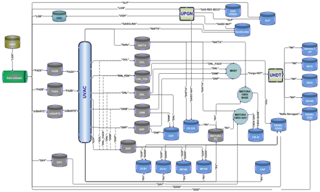
Objective: maximize profit

Variables:

- Stream Flows (crude, intermediate and final products);
- Storage;
- Stream properties.

Constraints

- Mass balance;
- Market features (supply and demand);
- Unit capacities;
- Stream property limits;
- Calculation of mix properties (nonlinear)



Stochastic Refinery Operations Planning Problem

The stochastic ROPP (SROPP) is an extension of ROPP to consider the uncertain nature of the oil supply.

Uncertainty: supply agreed in advance by long-term contracts that allows for unpredictable variability in

- Mix of oils;
- Quality of each oil;
- Quantity of each type.

Stochastic Refinery Operations Planning Problem

The stochastic ROPP (SROPP) is an extension of ROPP to consider the uncertain nature of the oil supply.

Uncertainty: supply agreed in advance by long-term contracts that allows for unpredictable variability in

- Mix of oils;
- Quality of each oil;
- Quantity of each type.

We use a two-stage stochastic programming model in which these uncertain quantities are represented by means of scenarios.

First-stage decision: oil acquired from spot market;
Second stage variables: flows, inventories, properties, *etc.*.

Outline of this talk

SROPP as a MIQCQP model

The SROPP has a special structure that we would like to explore by means of Lagrangian decomposition.

$$\begin{aligned}
\max \; & \sum_{s \in S} (x_s^T Q_0 x_s + f_0(x_s, y_s)) \\
\text{s.t.:} \; & x_s^T Q_r x_s + f_r(x_s, y_s) \leq 0, \; \forall s \in S, \forall r \in \mathcal{C}_s \equiv I_{1,m} \\
& x_{s,i} \in [X_{s,i}^L, X_{s,i}^U], \; \forall s \in S, \forall i \in \mathcal{VC}_s \equiv I_{1,n_1} \\
& y_{s,i} \in \{Y_{s,i}^L, \ldots, Y_{s,i}^U\}, \; \forall s \in S, \forall i \in \mathcal{VI}_s \equiv I_{1,n_2} \\
& \sum_{s \in S} (A_s^1 x_s + B_s^1 y_s) = b^1 \\
& \sum_{s \in S} (A_s^2 x_s + B_s^2 y_s) \geq b^2.
\end{aligned}$$

$I_{a,b} \subset \mathcal{Z} := \{a, \ldots, b\}$, $\forall r \in I_{0,m}$, Q_r is a symmetric matrix, f_0 is a linear function an f_r is an affine function $\forall r \in I_{1,m}$; $x \in [X^L, X^U]$ and $y \in \{Y^L, Y^U\}$; A_s, B_s are matrix and b a vector of adequate size.

SROPP as a MIQCQP model

The SROPP has a special structure that we would like to explore by means of Lagrangian decomposition.

$$
\begin{aligned}
\max \ & \sum_{s \in S} (x_s^T Q_0 x_s + f_0(x_s, y_s)) \\
\text{s.t.:} \ & x_s^T Q_r x_s + f_r(x_s, y_s) \le 0, \ \forall s \in S, \forall r \in \mathcal{C}_s \equiv I_{1,m} \\
& x_{s,i} \in [X_{s,i}^L, X_{s,i}^U], \ \forall s \in S, \forall i \in \mathcal{VC}_s \equiv I_{1,n_1} \\
& y_{s,i} \in \{Y_{s,i}^L, \dots, Y_{s,i}^U\}, \ \forall s \in S, \forall i \in \mathcal{VI}_s \equiv I_{1,n_2} \\
& \sum_{s \in S} (A_s^1 x_s + B_s^1 y_s) = b^1 \\
& \sum_{s \in S} (A_s^2 x_s + B_s^2 y_s) \ge b^2.
\end{aligned}
$$

$s \in S$ are separable subproblems; $\mathcal{C}_s$, $\mathcal{VC}_s$ and $\mathcal{VI}_s$ are constraints, continuous, and integer variables for $s \in S$. For $s_1, s_2 \in S$ such that $s_1 \neq s_2$, $\mathcal{C}_{s_1} \cap \mathcal{C}_{s_2} = \emptyset$, $\mathcal{VC}_{s_1} \cap \mathcal{VC}_{s_2} = \emptyset$ and $\mathcal{VI}_{s_1} \cap \mathcal{VI}_{s_2} = \emptyset$.

Lagrangian relaxation and MIQCQPs

Relaxing the complicating constraints, we obtain the completely separable Lagrangian dual subproblem (DSP)

$$\min_{\mu,\lambda} \phi(\mu,\lambda) = \sum_{s\in S} \phi_s(\mu,\lambda)$$

where $\phi_s(\mu,\lambda) :=$

$$\begin{aligned}
\max\ & (x_s^T Q_0 x_s + f_0(x_s, y_s)) - \mu^T(A_s^1 x_s + B_s^1 y_s) + \lambda^T(A_s^2 x_s + B_s^2 y_s) \\
\text{s.t.:}\ & x_s^T Q_r x_s + f_r(x_s, y_s) \leq 0, \quad \forall s \in S, \forall r \in \mathcal{C}_s \equiv I_{1,m} \\
& x_{s,i} \in [X_{s,i}^L, X_{s,i}^U], \quad \forall s \in S, \forall i \in \mathcal{VC}_s \equiv I_{1,n_1} \\
& y_{s,i} \in \{Y_{s,i}^L, \dots, Y_{s,i}^U\}, \quad \forall s \in S, \forall i \in \mathcal{VI}_s \equiv I_{1,n_2}.
\end{aligned}$$

This framework is however flawed due to the nonconvex nature of our MIQCQP...

Lagrangian relaxation and MIQCQPs

Despite DSP being suitable for non-smooth optimisation methods, it would require MIQCQPs to be solved optimally.

- Typically, a dual step obtains (μ, λ) (multiplier updates), while a primal step calculates $\phi(\mu, \lambda)$;
- Weak duality $z^* \leq \phi(\mu, \lambda)$ cannot be relied upon, as $\phi(\mu, \lambda)$ might not be solvable.

Lagrangian relaxation and MIQCQPs

Despite DSP being suitable for non-smooth optimisation methods, it would require MIQCQPs to be solved optimally.

- Typically, a dual step obtains (μ, λ) (multiplier updates), while a primal step calculates $\phi(\mu, \lambda)$;
- Weak duality $z^* \leq \phi(\mu, \lambda)$ cannot be relied upon, as $\phi(\mu, \lambda)$ might not be solvable.

To circumvent this issue, we take a different road, by replacing the MIQCQPs with arbitrarily good MIP relaxations.

- We replace $\phi(\mu, \lambda)$ with $\hat{\phi}_p(\mu, \lambda)$ such that $z^* \leq \phi(\mu, \lambda) \leq \hat{\phi}_p(\mu, \lambda)$ can be used as a proxy;
- $\hat{\phi}_p(\mu, \lambda)$ can be made as close to $\phi(\mu, \lambda)$ as desired, trading off computational burden.

Normalised multi-parametric disaggregation technique

The developments presented next are based on the works of Teles et al. [12], Kolodziej et al. [5], and Castro [4].

In what follows, we focus on the subproblem:

$$\begin{aligned}
\max\ & x^T Q_0 x + f_0(x,y) \\
\text{s.t.:}\ & x^T Q_r x + f_r(x,y) \leq 0 \quad \forall r \in I_{1,m} \\
& x_i \in [X_i^L, X_i^U] \quad \forall i \in I_{1,n_1} \\
& y_i \in \{Y_i^L, \dots, Y_i^U\} \quad \forall i \in I_{1,n_2}
\end{aligned}$$

Let $QT = \{(i,j) \in I_{1,n_1} \times I_{1,n_1} | j \geq i, \exists r \in I_{0,m}, |Q_{r,i,j}| > 0\}$ and $DS = \{j \in I_{1,n_1} | \exists i \in I_{1,n_1}, (i,j) \in QT\}$.

Normalised multi-parametric disaggregation technique

First, we normalise x_j, $j \in DS$, using variable $\lambda_j \in [0, 1]$ as

$$x_j = (X_j^U - X_j^L)\lambda_j + X_j^L \tag{1}$$

Normalised multi-parametric disaggregation technique

First, we normalise x_j, $j \in DS$, using variable $\lambda_j \in [0, 1]$ as

$$x_j = (X_j^U - X_j^L)\lambda_j + X_j^L \quad (1)$$

Variable λ_j is then discretised (binary-expanded with basis 10) as

$$\lambda_j = \sum_{k \in I_{0,9}, l \in I_{p,-1}} 10^l k z_{j,k,l} + \Delta\lambda_j \quad \forall j \in DS \quad (2)$$

$$\sum_{k \in I_{0,9}} z_{j,k,l} = 1 \quad \forall j \in DS, l \in I_{p,-1} \quad (3)$$

$$0 \leq \Delta\lambda_j \leq 10^p \quad \forall j \in DS \quad (4)$$

Notice that p is the parameter that controls the precision 10^{-p} of the relaxation (and of $\hat{\phi}_p(\mu, \lambda)$).

Normalised multi-parametric disaggregation technique

Multiplying equations (1) and (2) by x_i, $\forall i \in I_{1,n}$, we recover the product terms

$$x_i x_j = (X_j^U - X_j^L) x_i \lambda_j + x_i X_j^L \ \forall i,j \in QT$$
$$x_i \lambda_j = \sum_{k \in I_{0,9}, l \in I p, -1} 10^l k x_i z_{j,k,l} + x_i \Delta\lambda_j, \ \forall i,j \in QT$$

Normalised multi-parametric disaggregation technique

Multiplying equations (1) and (2) by x_i, $\forall i \in I_{1,n}$, we recover the product terms

$$x_i x_j = (X_j^U - X_j^L) x_i \lambda_j + x_i X_j^L \ \forall i, j \in QT$$
$$x_i \lambda_j = \sum_{k \in I_{0,9}, l \in I p, -1} 10^l k x_i z_{j,k,l} + x_i \Delta\lambda_j, \ \forall i, j \in QT$$

Using auxiliary variables $w_{i,j}$, $\hat{x}_{i,j,k,l}$, $v_{i,j}$, and $\Delta v_{i,j}$ to represent $x_i x_j$, $x_i z_{j,k,l}$, $x_i \lambda_j$, and $x_i \Delta\lambda_j$ (resp.), we obtain

$$w_{i,j} = (X_j^U - X_j^L) v_{i,j} + x_i X_j^L, \quad \forall i, j \in QT \tag{5}$$
$$v_{i,j} = \sum_{k \in I_{0,9}, l \in I_{p,-1}} k 10^l \hat{x}_{i,j,k,l} + \Delta v_{i,j}, \quad \forall i, j \in QT. \tag{6}$$

Normalised multi-parametric disaggregation technique

(7)–(9) - McCormick envelopes [7] that provide a relaxation of the product of two continuous variables;

$$X_i^L \Delta\lambda_j \leq \Delta v_{i,j} \leq X_i^U \Delta\lambda_j, \quad \forall i,j \in QT \tag{7}$$

$$10^p(x_i - X_i^U) + X_i^U \Delta\lambda_j \leq \Delta v_{i,j}, \ \forall i,j \in QT \tag{8}$$

$$10^p(x_i - X_i^L) + X_i^L \Delta\lambda_j \geq \Delta v_{i,j}, \ \forall i,j \in QT \tag{9}$$

(10)–(12) - define a exact linearisation.

$$\sum_{k \in I_{0,9}} z_{j,k,l} = 1, \quad \forall j \in DS, l \in I_{p,-1} \tag{10}$$

$$\sum_{k \in I_{0,9}} \hat{x}_{i,j,k,l} = x_i, \quad \forall i,j \in QT \tag{11}$$

$$X_i^L z_{j,k,l} \leq \hat{x}_{i,j,k,l} \leq X_i^U z_{j,k,l}, \quad \forall i,j,k,l. \tag{12}$$

Reformulated NMDT

In previous work[1], we proposed several enhancements that made this relaxation more efficient:

1. Use base 2 instead of base 10 for the discretisation;
2. Variable and constraint elimination;
3. Selection criteria of variables to have discretisation precision p increased (relevant for the decomposition method).

$$f_{\text{rank}}(j) = \sum_{r} |Q_{r,j}(w_{j,j} - x_j^2)| + 2 \sum_{((r,i)|i>j|(i,j)\in QT)} |Q_{r,i,j}(w_{i,j} - x_i x_j)|$$

These enhancements were paramount for the success of the p-Lagrangian method...

[1] In Andrade et. al; Enhancing the normalized multiparametric disaggregation technique for mixed-integer quadratic programming, to appear in *JOGO* (2018).

Reformulated NMDT (for completeness)

$$
\begin{aligned}
\max \quad & \sum_{i|(i,i)\in QT} Q_{0,i,i} w_{i,i} + 2 \sum_{(i,j)\in QT|j>i} Q_{0,i,j} w_{i,j} + f_0(x,y) \\
\text{s.t.:} \quad & \sum_{i|(i,i)\in QT} Q_{r,i,i} w_{i,i} + 2 \sum_{(i,j)\in QT|j>i} Q_{r,i,j} w_{i,j} + f_r(x,y) \leq 0, \quad \forall r \in I_{1,m} \\
& x_j = (X_j^U - X_j^L)\Big(\sum_{l\in I_{p,-1}} 2^l z_{j,l} + \Delta x_j\Big), \quad \forall j \in DS \\
& w_{i,j} = (X_j^U - X_j^L)\Big(\sum_{l\in I_{p,-1}} 2^l \hat{x}_{i,j,l} + \Delta w_{i,j}\Big), \quad \forall i \in I_{1,n}, j \in I_{1,n} | (i,j) \in QT \\
& 0 \leq \Delta\lambda_j \leq 2^p, \quad \forall j \in DS \\
& 2^p(x_i - X_i^U) + X_i^U \Delta x_j \leq \Delta v_{i,j} \leq 2^p(x_i - X_i^L) + X_i^L \Delta x_j, \quad \forall i,j|(i,j) \in QT \\
& x_i^L \Delta x_j \leq \Delta v_{i,j} \leq x_i^U \Delta x_j, \quad \forall i,j|(i,j) \in QT \\
& X_i^L z_{j,l} \leq \hat{x}_{i,j,l} \leq X_i^U z_{j,l}, \quad \forall i,j,l \in QT \times I_{0,p} \\
& X_i^L(1 - z_{j,l}) \leq x_i - \hat{x}_{i,j,l} \leq X_i^U(1 - z_{j,l}), \quad \forall i,j,l \in QT \times I_{0,p} \\
& x_i \in [X_i^L, X_i^U], \quad \forall i \in I_{1,n_1} \\
& y_i \in \{Y_i^L, \dots, Y_i^U\}, \quad \forall i \in I_{1,n_2} \\
& z_{j,l} \in \{0,1\}, \quad \forall j,l \in DS \times I_{0,p}.
\end{aligned}
$$

Outline of this talk

Introduction

(R)NMDT and MILP relaxation

p-Lagrangian decomposition

Numerical results

Conclusions and future directions

p-Lagrangian decomposition method

We have all ingredients to pose the algorithm. An informal pseudocode could be:

Algorithm 1 p-LD algorithm

Step 0. Set $p = 0$ and initial (μ, λ)
Step 1. Compute p-Lagrangian dual bound $\hat{\phi}_p(\mu, \lambda)$; obtain a feasible solution using a Lagrangian heuristic (if possible).
Step 2. Update (μ, λ) using your favourite nonsmooth optimization method;
Step 3. If a "dual" stopping condition is met, set $p = p - 1$ and return to Step 1;
Step 4. If a "primal" stopping condition is met, stop. Otherwise, return to Step 1.

Outline of this talk

Instances generated

To test the framework, we generated random instances replicating the SROPP by:

1. Generating $|S|$ subproblems with random $f_r(x, y)$, $r \in I_{i,m}$;
2. Enforcing constraints $x_s = x_{s+1}$, $s \in S \setminus \{|S|\}$ that simulates nonaticipativity conditions;
3. Artificially creating dense subproblems (fully dense linear and quadratic terms.

Instances generated

To test the framework, we generated random instances replicating the SROPP by:

1. Generating $|S|$ subproblems with random $f_r(x, y)$, $r \in I_{i,m}$;
2. Enforcing constraints $x_s = x_{s+1}$, $s \in S \setminus \{|S|\}$ that simulates nonaticipativity conditions;
3. Artificially creating dense subproblems (fully dense linear and quadratic terms.

The algorithmic setup: RNMDT + Bundle method; Primal solutions: fix integer values and solve an NLP (CONOPT). Computer: Intel i7 16GB. Implemented using GAMS.

We compare the performance of our methods with: RNMDT-only and with the global solvers: Lindo Global [6], SCIP [1], Couenne [3], Antigone [8] and Baron [11] using the NEOS Server.

Instances generated

continuous				mixed-integer
instance	$\lvert S \rvert$	$\lvert \mathcal{VC} \rvert$	$\lvert \mathcal{C} \rvert$	$\lvert \mathcal{VI} \rvert$
1	2	21	41	8
2	3	31	61	12
3	4	41	81	16
4	5	51	101	20
5	6	61	121	24
6	7	71	141	28
7	8	81	161	32
8	9	91	181	36
9	10	101	201	40
10	20	201	401	80
11	30	301	601	120
12	40	401	801	160
13	50	501	1001	200
14	60	601	1201	240
15	70	701	1401	280
16	80	801	1601	320
17	90	901	1801	260
18	100	1001	2001	400

Computational results

instance	RNMDT	p-LR	LIN.G.	SCIP	COU.	ANT.	BAR.
1	0.0%	0.0%	153.6%	68.3%	0.0%	0.0%	0.0%
2	1.7%	0.0%	269.2%	201.7%	0.0%	3.9%	0.0%
3	8.5%	0.0%	267.5%	255.2%	14.9%	8.1%	4.4%
4	14.4%	0.0%	295.3%	275.7%	30.1%	9.4%	4.9%
5	17.2%	0.0%	298.1%	287.7%	40.0%	10.2%	6.2%
6	23.3%	0.0%	324.0%	300.3%	459.0%	13.5%	10.6%
7	27.2%	0.0%	306.8%	308.5%	57.1%	13.1%	11.4%
8	27.5%	0.0%	330.3%	322.0%	53.6%	13.5%	11.2%
9	28.5%	0.0%	325.9%	304.5%	170.0%	16.5%	15.7%
10	41.8%	0.0%	350.5%	362.0%	76.6%	18.8%	21.1%
11	47.5%	0.0%	395.1%	367.6%	83.3%	24.2%	24.6%
12	48.9%	0.0%	369.5%	378.9%	115.5%	21.2%	24.0%
13	52.1%	0.0%	498.3%	358.5%	85.1%	24.0%	27.8%
14	54.0%	0.0%	486.7%	373.8%	89.0%	24.7%	29.3%
15	52.8%	0.0%	504.6%	360.7%	89.0%	23.3%	30.2%
16	53.3%	0.0%	507.4%	374.7%	94.9%	24.9%	31.5%
17	55.9%	0.0%	500.9%	368.1%	97.3%	25.4%	33.9%
18	56.7%	0.0%	#N/A	358.4%	110.2%	24.8%	34.3%

Table: Continuous instances - relative gap

Computational results

instance	RNMDT	p-LR	LIN.G.	SCIP	COU.	ANT.	BAR.
1	23.2%	0.0%	290.5%	93.8%	0.0%	0.0%	0.0%
2	25.0%	0.0%	326.7%	182.5%	19.4%	5.8%	0.0%
3	29.9%	0.0%	327.2%	226.5%	239.4%	9.2%	5.7%
4	38.9%	0.0%	337.9%	271.2%	63.2%	11.4%	9.5%
5	45.2%	0.0%	331.8%	283.8%	74.1%	14.9%	13.5%
6	48.2%	0.0%	328.0%	281.1%	135.5%	14.8%	14.0%
7	48.5%	0.0%	334.8%	319.8%	75.5%	15.4%	16.2%
8	62.5%	0.0%	346.5%	316.7%	86.2%	18.1%	18.1%
9	65.5%	0.3%	421.9%	317.6%	85.1%	19.6%	20.1%
10	67.4%	0.0%	503.7%	365.8%	88.8%	19.3%	21.0%
11	85.5%	0.0%	436.8%	359.5%	82.4%	23.7%	27.8%
12	#N/A	0.0%	519.8%	331.1%	91.0%	23.6%	25.6%
13	#N/A	0.0%	709.7%	358.7%	85.2%	24.3%	29.0%
14	#N/A	0.0%	#N/A	358.3%	91.7%	24.8%	31.7%
15	#N/A	0.0%	6552.8%	355.7%	80.3%	24.5%	29.8%
16	124.6%	0.0%	#N/A	357.5%	87.1%	24.3%	32.7%
17	#N/A	0.0%	#N/A	361.6%	82.0%	25.9%	33.8%
18	#N/A	0.0%	#N/A	359.2%	81.8%	26.9%	32.6%

Table: Mixed integer instances - relative gap

Computational results

instance	RNMDT	p-LR	LIN.G.	SCIP	COU.	ANT.	BAR.
1	1.27E+02	2.29E+01	T	T	6.30E+01	1.77E+02	1.60E+01
2	T	4.45E+01	T	T	2.10E+03	T	2.20E+03
3	T	1.27E+02	T	T	T	T	T
4	T	1.27E+02	T	T	T	T	T
5	T	7.66E+01	T	T	T	T	T
6	T	1.93E+02	T	T	T	T	T
7	T	1.37E+02	T	T	T	T	T
8	T	2.13E+02	T	T	T	T	T
9	T	2.18E+02	T	T	T	T	T
10	T	3.70E+03	T	T	T	T	T
11	T	7.35E+02	T	T	T	T	T
12	T	4.10E+03	T	T	T	T	T
13	T	1.50E+03	T	T	T	T	T
14	T	2.37E+03	T	T	T	T	T
15	T	2.11E+03	T	T	T	T	T
16	T	T	T	T	T	T	T
17	T	4.17E+03	T	T	T	T	T
18	T	6.16E+03	T	T	T	T	T

Table: Continuous instances - time in s (T - timed out 2h)

Computational results

instance	RNMDT	p-LR	LIN.G.	SCIP	COU.	ANT.	BAR.
1	T	3.76E+01	T	T	2.43E+02	5.42E+02	6.20E+01
2	T	5.51E+01	T	T	T	T	2.98E+03
3	T	5.94E+01	T	T	T	T	T
4	T	4.92E+01	T	T	T	T	T
5	T	6.70E+01	T	T	T	T	T
6	T	6.36E+01	T	T	T	T	T
7	T	1.20E+02	T	T	T	T	T
8	T	1.00E+02	T	T	T	T	T
9	T	T	T	T	T	T	T
10	T	1.25E+03	T	T	T	T	T
11	T	1.17E+03	T	T	T	T	T
12	T	2.15E+03	T	T	T	T	T
13	T	2.08E+03	T	T	T	T	T
14	T	2.29E+03	T	T	T	T	T
15	T	4.85E+03	T	T	T	T	T
16	T	3.84E+03	T	T	T	T	T
17	T	3.28E+03	T	T	T	T	T
18	T	4.33E+03	T	T	T	T	T

Table: Continuous instances - time in s (T - timed out 2h)

Outline of this talk

Conclusions and future directions

Main insights:

- The combination of RNMDT within a decomposition framework is the key reason of the performance improvement;
- The idea of using p-Lagrangian relaxation opens up an interesting avenue for tackling large-scale (MI)QCQPs.

Conclusions and future directions

Main insights:

- The combination of RNMDT within a decomposition framework is the key reason of the performance improvement;
- The idea of using p-Lagrangian relaxation opens up an interesting avenue for tackling large-scale (MI)QCQPs.

Directions for future research include:

1. Adapt this for adaptive robust optimisation problems with bilevel structure;
2. Work on a general implementation for RNMDT that can be reused in multiple contexts;
3. Investigate parallelisation strategies from SMIP recent advances.

The p-Lagrangian method for MIQCQPs

Fabricio Oliveira

Department of Mathematics and Systems Analysis
School of Science, Aalto University

ISMP 2018 - Bordeaux, France

July 5, 2018

Aalto University
School of Science

PUC RIO

RMIT UNIVERSITY

Additional references I

[1] Tobias Achterberg. Scip: solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

[2] Tiago Andrade, Gabriela Ribas, and Fabricio Oliveira. A strategy based on convex relaxation for solving the oil refinery operations planning problem. *Industrial & Engineering Chemistry Research*, 55(1):144–155, 2016.

[3] Pietro Belotti. Couenne: a users manual. Technical report, Technical report, Lehigh University, 2009.

[4] Pedro M Castro. Normalized multiparametric disaggregation: an efficient relaxation for mixed-integer bilinear problems. *Journal of Global Optimization*, 64(4):765–784, 2016.

[5] Scott Kolodziej, Pedro M Castro, and Ignacio E Grossmann. Global optimization of bilinear programs with a multiparametric disaggregation technique. *Journal of Global Optimization*, 57(4):1039–1063, 2013.

[6] Youdong Lin and Linus Schrage. The global solver in the lindo api. *Optimization Methods & Software*, 24(4-5):657–668, 2009.

[7] Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part I convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

Additional references II

[8] Ruth Misener and Christodoulos A Floudas. Antigone: algorithms for continuous/integer global optimization of nonlinear equations. *Journal of Global Optimization*, 59(2-3):503–526, 2014.

[9] Sérgio MS Neiro and José M Pinto. A general modeling framework for the operational planning of petroleum supply chains. *Computers & Chemical Engineering*, 28(6):871–896, 2004.

[10] Sérgio MS Neiro and José M Pinto. Multiperiod optimization for production planning of petroleum refineries. *Chem. Eng. Comm.*, 192(1):62–88, 2005.

[11] Nikolaos V Sahinidis. Baron: A general purpose global optimization software package. *Journal of global optimization*, 8(2):201–205, 1996.

[12] João P Teles, Pedro M Castro, and Henrique A Matos. Global optimization of water networks design using multiparametric disaggregation. *Computers & Chemical Engineering*, 40:132–147, 2012.