Mixed-Integer Nonlinear Optimisation: Reformulations & Special Structure Recognition

Ruth Misener
r.misener@imperial.ac.uk

Computational Optimisation Group
Centre for Process Systems Engineering
Department of Computing

Imperial College London

computational optimisation group

Centre for Process Systems Engineering

2017/05/17

Relaxations of bilinear functions (Tawarmalani et al., 2002)

$$\phi(y, x_1, \ldots, x_n) = b_0 y + y \cdot \sum_{k=1}^{n} b_k x_k - a_0 - \sum_{k=1}^{n} a_k x_k$$

Key Idea - Product disaggregation

Distribute the product y over the sum $\sum_{k=1}^{n} b_k x_k$ and then bound the resulting n bilinear terms.

Convex envelope of a bilinear function [1/2]

Tawarmalani et al., 2002

$$\phi(y, x_1, \ldots, x_n) = b_0 y + y \cdot \sum_{k=1}^{n} b_k x_k - a_0 - \sum_{k=1}^{n} a_k x_k$$

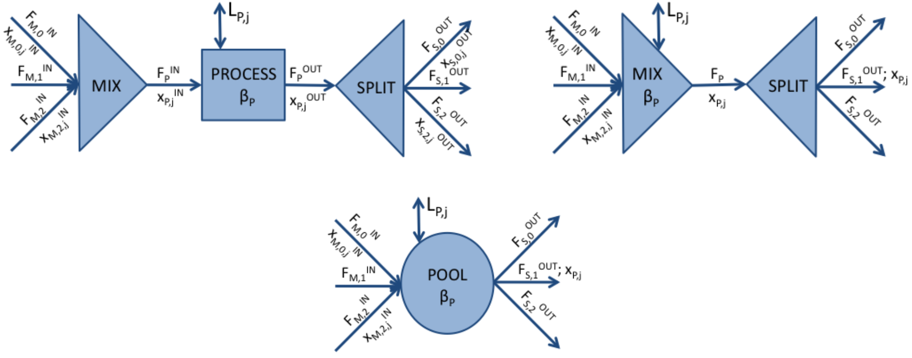
Assume $x_k \in [x_k^L, x_k^U], y \in [y^L, y^U]$. Let $H^{n+1} = \prod_{k=1}^{n}[x_k^L, x_k^U] \times [y^L, y^U]$.
Then:

$$\begin{aligned}
&\operatorname{convenv}_{H^{n+1}} \phi(y, x_1, \ldots, x_n) \\
= &\operatorname{convenv}_{H^{n+1}} \left(b_0 y + y \cdot \sum_{k=1}^{n} b_k x_k - a_0 - \sum_{k=1}^{n} a_k x_k \right) \\
&\qquad = b_0 y + \sum_{k=1}^{n} \operatorname{convenv}_{[x_k^L, x_k^U] \times [y^L, y^U]} \left(b_k y \cdot x_k \right) - a_0 - \sum_{k=1}^{n} a_k x_k
\end{aligned}$$

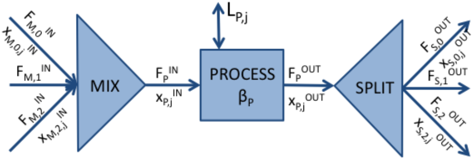
Convex envelope of a bilinear function [2/2]

Why? For fixed $\mathbf{x}$, $\phi(y, \mathbf{x})$ is linear in f. Let Φ be the epigraph of the convex envelope of $\phi(y, \mathbf{x})$. Then Φ can be expressed as the convex hull of $A = \left\{(\phi^a, \mathbf{x}^a) | \phi^a \geq \phi(\mathbf{x}^a, y^L)\right\}$ and $B = \left\{(\phi^b, \mathbf{x}^b) | \phi^b \geq \phi(\mathbf{x}^b, y^U)\right\}$, i.e.:

Need for reformulation [1/2]



Need for reformulation [2/2]



$$\text{Mixer} \quad \begin{cases} \sum_i F_{M,i} = F_T^{IN} \\ \sum_i x_{M,i,j} \cdot F_{M,i} = x_{T,j}^{IN} \cdot F_T^{IN} \quad \forall\, j \in \{1, \ldots, J\} \end{cases}$$

$$\text{Treatment} \quad \begin{cases} F_T^{IN} = F_T^{OUT} \\ x_{T,j}^{IN} = \beta_{T,j} \cdot x_{T,j}^{OUT} \quad \forall\, j \in \{1, \ldots, J\} \end{cases}$$

$$\text{Splitter} \quad \begin{cases} F_T^{OUT} = \sum_i F_{S,i} \\ x_{T,j}^{OUT} = x_{T,i,j} \quad \forall\, i \in \{1, \ldots, I\};\ j \in \{1, \ldots, J\} \end{cases}$$

where the x and F variables represent concentrations and flowrates, the i and j indices represent individual flowrates and monitored quality components, and the parameter $\beta_{T,j}$ represents the removal of undesired quality j.

Eliminating Variables [1/2]

1. For linear equality constraints with continuous variables x_i, x_j and coefficient parameters $a_i, a_j \in \mathbb{R}$:

$$a_i \cdot x_i + a_j \cdot x_j = b$$

   GloMIQO eliminates x_i from the optimization problem by substituting $\frac{b - a_j \cdot x_j}{a_i}$ for every instantiation of variable x_i and, if variable x_i had finite bounds, adding a constraint $x_i^{\mathrm{LO}} \leq \frac{b - a_j \cdot x_j}{a_i} \leq x_i^{\mathrm{UP}}$.

2. For linear equality constraints with two binary variables y_i, y_j and coefficient parameters $c_i, c_j \in \mathbb{R}$:

$$c_i \cdot y_i + c_j \cdot y_j = b$$

   GloMIQO eliminates y_i from the optimization problem by substituting $\frac{b - c_j \cdot y_j}{c_i}$ for every instantiation of variable y_i and adding a constraint $y_i^{\mathrm{LO}} \leq \frac{b - c_j \cdot y_j}{c_i} \leq y_i^{\mathrm{UP}}$.

Eliminating Variables [2/2]

3. For linear equality constraints with continuous variable x_i that participates linearly in MIQCQP, binary variable y_j and coefficient parameters $a_i, c_j \in \mathbb{R}$:

$$a_i \cdot x_i + c_j \cdot y_j = b$$

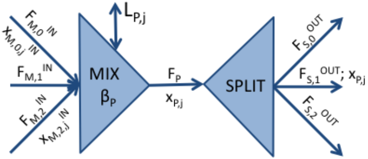
   GloMIQO eliminates x_i from the optimization problem by substituting $\frac{b - c_j \cdot y_j}{a_i}$ for every instantiation of variable x_i and, if variable x_i had finite bounds, adding a constraint $x_i^{\mathrm{LO}} \leq \frac{b - c_j \cdot y_j}{a_i} \leq x_i^{\mathrm{UP}}$.

4. For equality constraints with bilinear or quadratic term $x_i \cdot x_j$, continuous variable x_k that participates linearly in MIQCQP, and coefficient parameters $q_{ij}, a_k \in \mathbb{R}$:

$$q_{ij} \cdot x_i \cdot x_j + a_k \cdot x_k = b$$

   GloMIQO eliminates x_k from the optimization problem by substituting $\frac{b - q_{ij} \cdot x_i \cdot x_j}{a_k}$ for every instantiation of variable x_k, if variable x_k had finite bounds, and adding a constraint $x_k^{\mathrm{LO}} \leq \frac{b - q_{ij} \cdot x_i \cdot x_j}{a_k} \leq x_k^{\mathrm{UP}}$.

Outcome of eliminating variables



$$\text{Mixer} \quad \begin{cases} \sum_i F_{M,i} = F_T^{IN} \\ \sum_{i,j} x_{M,i,j} \cdot F_{M,i} = \beta_{T,j} \cdot x_{T,j}^{OUT} \cdot F_T^{IN} \quad \forall\, j \in \{1, \ldots, J\} \end{cases}$$

$$\text{Splitter} \quad \begin{cases} F_T^{IN} = \sum_i F_{S,i} \end{cases}$$

Disaggregating Bilinear Terms [1/2]

1. For linear equality constraints with one continuous variable x_i that participates nonlinearly elsewhere in the MIQCQP optimization problem and J continuous variables x_j that exclusively participate linearly in MIQCQP:

$$a_i \cdot x_i + \sum_{j=1}^{J} a_j \cdot x_j = b$$

GloMIQO eliminates x_i from the optimization problem by substituting $\frac{b - \sum_{j=1}^{J} a_j \cdot x_j}{a_i}$ for every instantiation of variable x_i and, if variable x_i had finite bounds, adding a constraint $x_i^{\mathrm{LO}} \leq \frac{b - \sum_{j=1}^{J} a_j \cdot x_j}{a_i} \leq x_i^{\mathrm{UP}}$.

Disaggregating Bilinear Terms [1/2]

2. For linear equality constraints with I continuous variables x_i that are bounded by 0 (*i.e.*, $x_i^{\mathrm{LO}} \geq 0 \quad \forall\, i \in \{1, \ldots, I\}$):

$$\sum_{i=1}^{I} a_i \cdot x_i = b$$

and have coefficient parameters a_i with:

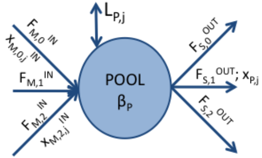
$$a_1 > 0 \text{ and } a_i < 0 \quad \forall\, i \in \{2, \ldots, I\}$$

or, symmetrically:

$$a_1 < 0 \text{ and } a_i > 0 \quad \forall\, i \in \{2, \ldots, I\}$$

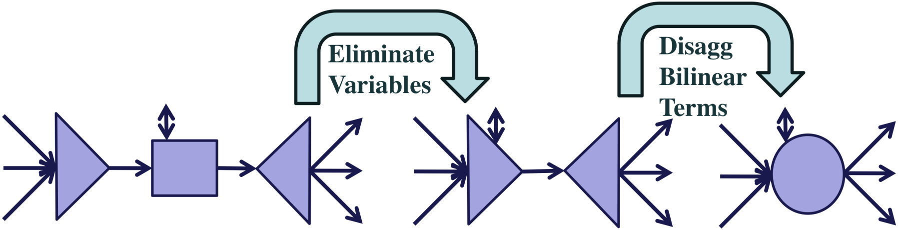
GloMIQO eliminates x_1 from the optimization problem by substituting $\frac{b - \sum_{i=2}^{I} a_i \cdot x_i}{a_1}$ for every instantiation of variable x_1 and, if variable x_1 had finite bounds, adding a constraint $x_1^{\mathrm{LO}} \leq \frac{b - \sum_{i=2}^{I} a_i \cdot x_i}{a_1} \leq x_1^{\mathrm{UP}}$.

Result of the reformulation



$$\text{Pool} \quad \begin{cases} \sum_{i,j} x_{M,i,j} \cdot F_{M,i} = \sum_i \beta_{T,j} \cdot x_{T,j}^{OUT} \cdot F_{M,i} \quad \forall j \in \{1, \ldots, J\} \\ \sum_i F_{M,i} = \sum_i F_{S,i} \end{cases}$$

waste - MINLPLib



GAMS Development, GAMS Software Client Models

MINLPLib **waste**:

- **2084** continuous variables
- **400** binary variables
- **1992** equations
- **1368** bilinear terms

GloMIQO reformulation:

- **666** continuous variables
- **400** binary variables
- **1980** equations (added **72**)
- **1284** bilinear terms