A Simple Heuristic Based on Alternating Direction Method of Multipliers for Solving Mixed-Integer Nonlinear Optimization

Yoshihiro Kanno[†] Satoshi Kitayama[‡]

[†]The University of Tokyo

[‡]Kanazawa University

May 21–24, 2018 (ACSMO 2018)

a challenging optim. prob.

- mixed-integer nonlinear programming (MINLP):

$$
\begin{aligned}
\text{Min.} \quad & f(\boldsymbol{x}) \\
\text{s. t.} \quad & g_i(\boldsymbol{x}) \le 0 \quad (i = 1, \dots, m), \\
& x_1, \dots, x_r \quad \text{: discrete,} \\
& x_{r+1}, \dots, x_n \quad \text{: continuous.}
\end{aligned}
$$

  - hard to solve

a challenging optim. prob.

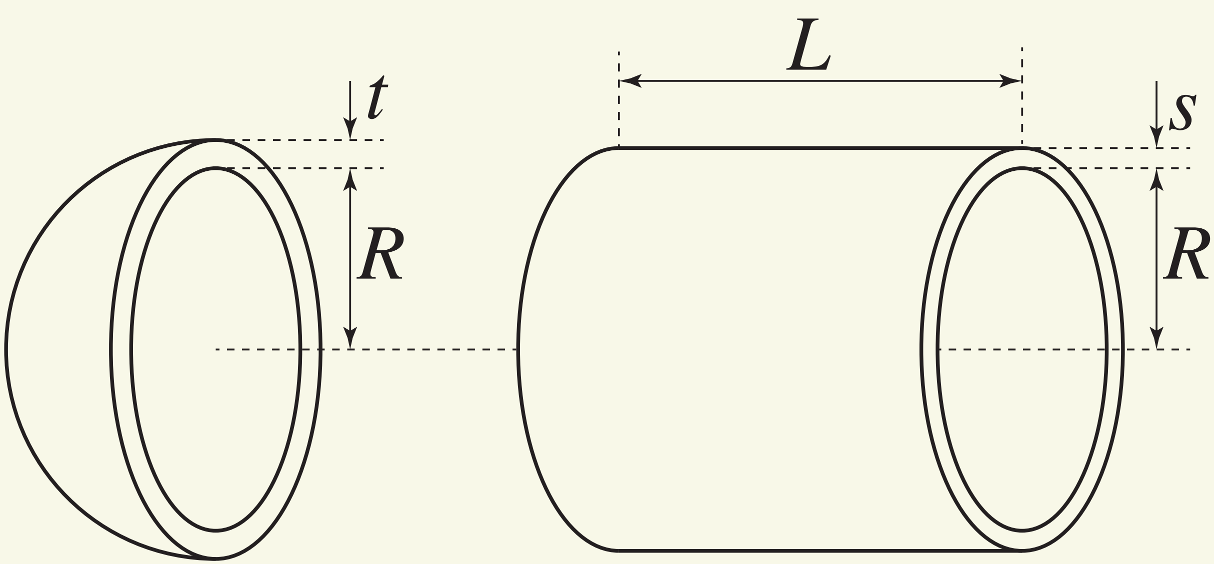
- mixed-integer nonlinear programming (MINLP):

$$
\begin{array}{lll}
\text{Min.} & f(\boldsymbol{x}) & \\
\text{s. t.} & g_i(\boldsymbol{x}) \le 0 & (i = 1, \dots, m), \\
 & x_1, \dots, x_r & : \text{discrete}, \\
 & x_{r+1}, \dots, x_n & : \text{continuous}.
\end{array}
$$

  - hard to solve
- our heuristic
  - simple
  - computationally cheap
  - often finds good solutions
- metaheuristics
  - generally, computationally expensive

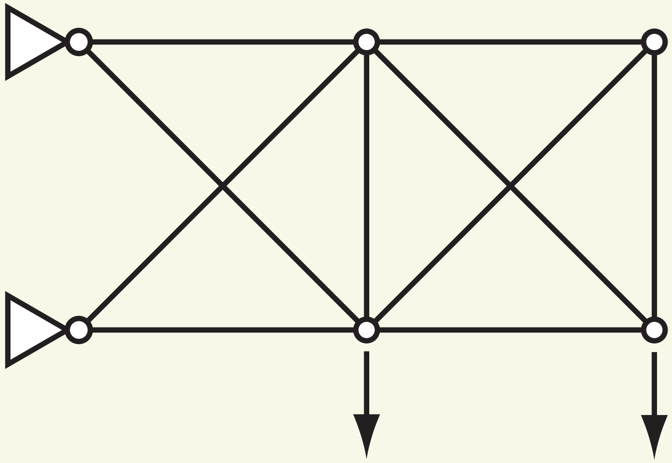
ex. of MINLP (1/2)

- optimal design of pressure vessel [Sandgren '90]
  - 4 design variables:
    - s, t : discrete (chosen from catalog)
    - L, R : continuous
  - objective: min. manufacturing cost (nonconvex)
  - constraints: ASME code (nonconvex)



ex. of MINLP (2/2)

- optimal design of truss w/ discrete cross-sections
  - design variables:
    - member cross-sectional areas : discrete (from catalog)
    - state variables (e.g., displacement) : continuous
  - objective: min. structural weight (topology is fixed)
  - constraints: stress & displacement
    - equilibrium eq. (nonconvex)



basic idea of our heuristic

- multiple restart of ADMM
  - from random initial points

basic idea of our heuristic

- multiple restart of ADMM
  - from random initial points
- ADMM = alternating direction method of multipliers
  - each iteration consists of...
    - solving an NLP (nonlinear programming) problem
    - rounding a vector to the nearest integer vector
    - updating dual variable (by vector addition)
  - often converges w/n 20 iterations
  - $\rightarrow$ simple & computationally cheap

ADMM = alternating direction method of multipliers

- an algorithm for convex optimization:

$$\begin{aligned} \text{Min.} \quad & f(\boldsymbol{x}) + g(\boldsymbol{z}) \\ \text{s. t.} \quad & A\boldsymbol{x} + B\boldsymbol{z} = \boldsymbol{c}. \end{aligned}$$

  - f, g : convex $\quad \boldsymbol{x}, \boldsymbol{z}$: variables

ADMM = alternating direction method of multipliers

- an algorithm for convex optimization:

$$\begin{array}{ll} \text{Min.} & f(\boldsymbol{x}) + g(\boldsymbol{z}) \\ \text{s. t.} & A\boldsymbol{x} + B\boldsymbol{z} = \boldsymbol{c}. \end{array}$$

  - f, g : convex $\boldsymbol{x}, \boldsymbol{z}$: variables

- classically, [Glowinski & Marrocco '75], [Gabay & Mercier '76]
  - distributed optim. for convex prob.
  - a precursor: method of multipliers

- recently,
  - heuristic for nonconvex prob.
    - data science [Chartrand '07], [Kanamori & Takeda '14]
    - mixed-integer convex program [Takapoui, Moehle, Boyd, & Bemporad '18]

iteration of ADMM

- target (f, g : convex):

$$\begin{aligned} \text{Min.} \quad & f(\boldsymbol{x}) + g(\boldsymbol{z}) \\ \text{s. t.} \quad & A\boldsymbol{x} + B\boldsymbol{z} = \boldsymbol{c}. \end{aligned}$$

- augmented Lagrangian:

$$\begin{aligned} L_\rho(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{y}) = & f(\boldsymbol{x}) + g(\boldsymbol{z}) \\ & + \boldsymbol{y}^\top (A\boldsymbol{x} + B\boldsymbol{z} - \boldsymbol{c}) + \frac{\rho}{2} \|A\boldsymbol{x} + B\boldsymbol{z} - \boldsymbol{c}\|^2 \end{aligned}$$

  - $\boldsymbol{y}$: Lagrange multiplier $\quad$ $\rho > 0$: penalty parameter

iteration of ADMM

- target (f, g : convex):

$$\begin{aligned} \text{Min.} \quad & f(\boldsymbol{x}) + g(\boldsymbol{z}) \\ \text{s. t.} \quad & A\boldsymbol{x} + B\boldsymbol{z} = \boldsymbol{c}. \end{aligned}$$

- augmented Lagrangian:

$$\begin{aligned} L_\rho(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{y}) = & f(\boldsymbol{x}) + g(\boldsymbol{z}) \\ & + \boldsymbol{y}^\top (A\boldsymbol{x} + B\boldsymbol{z} - \boldsymbol{c}) + \frac{\rho}{2} \| A\boldsymbol{x} + B\boldsymbol{z} - \boldsymbol{c} \|^2 \end{aligned}$$

  - $\boldsymbol{y}$: Lagrange multiplier $\quad$ $\rho > 0$: penalty parameter

- ADMM (primal variables are updated alternately):

$$\boldsymbol{x}^{k+1} \leftarrow \text{minimizer of } L_\rho(\boldsymbol{x}, \boldsymbol{z}^k; \boldsymbol{y}^k), \qquad \text{(primal update-1)}$$

$$\boldsymbol{z}^{k+1} \leftarrow \text{minimizer of } L_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{z}; \boldsymbol{y}^k), \qquad \text{(primal update-2)}$$

$$\boldsymbol{y}^{k+1} \leftarrow \boldsymbol{y}^k + \rho(A\boldsymbol{x}^{k+1} + B\boldsymbol{z}^{k+1} - \boldsymbol{c}). \qquad \text{(dual update)}$$

iteration of ADMM

- target (f, g : convex):

$$\begin{aligned} \text{Min.} \quad & f(\boldsymbol{x}) + g(\boldsymbol{z}) \\ \text{s. t.} \quad & A\boldsymbol{x} + B\boldsymbol{z} = \boldsymbol{c}. \end{aligned}$$

- augmented Lagrangian:

$$\begin{aligned} L_\rho(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{y}) = & f(\boldsymbol{x}) + g(\boldsymbol{z}) \\ & + \boldsymbol{y}^\top (A\boldsymbol{x} + B\boldsymbol{z} - \boldsymbol{c}) + \frac{\rho}{2}\|A\boldsymbol{x} + B\boldsymbol{z} - \boldsymbol{c}\|^2 \end{aligned}$$

  - $\boldsymbol{y}$: Lagrange multiplier $\qquad \rho > 0$: penalty parameter

- method of multipliers (= precursor of ADMM):

$$\begin{aligned} (\boldsymbol{x}^{k+1}, \boldsymbol{z}^{k+1}) & \leftarrow \text{minimizer of } L_\rho(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{y}^k), && \text{(primal update)} \\ \boldsymbol{y}^{k+1} & \leftarrow \boldsymbol{y}^k + \rho(A\boldsymbol{x}^{k+1} + B\boldsymbol{z}^{k+1} - \boldsymbol{c}). && \text{(dual update)} \end{aligned}$$

our method (1/3)

- discrete variables only (for simple presentation):

$$
\begin{aligned}
\text{Min.} \quad & f(\boldsymbol{x}) \\
\text{s. t.} \quad & g_i(\boldsymbol{x}) \leq 0 \quad (i = 1, \dots, m), && (\to \text{write } \boldsymbol{x} \in \mathcal{G}) \\
& x_1, \dots, x_n \quad : \text{from a list.} && (\to \text{write } \boldsymbol{x} \in \mathcal{D})
\end{aligned}
$$

our method (1/3)

- discrete variables only (for simple presentation):

$$\begin{aligned} \text{Min.} \quad & f(\boldsymbol{x}) \\ \text{s. t.} \quad & g_i(\boldsymbol{x}) \le 0 \quad (i = 1, \dots, m), && (\rightarrow \text{write } \boldsymbol{x} \in \mathcal{G}) \\ & x_1, \dots, x_n \quad : \text{from a list.} && (\rightarrow \text{write } \boldsymbol{x} \in \mathcal{D}) \end{aligned}$$

- reformulation:

$$\begin{aligned} \text{Min.} \quad f(\boldsymbol{x}) \quad \text{s. t.} \quad & \boldsymbol{x} = \boldsymbol{z} && (♠\text{-1}) \\ & \boldsymbol{x} \in \mathcal{G}, \quad \boldsymbol{z} \in \mathcal{D}. && (♠\text{-2}) \end{aligned}$$

- augmented Lagrangian:

$$L_\rho(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{y}) = f(\boldsymbol{x}) + \boldsymbol{y}^\top(\boldsymbol{x} - \boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{z}\|^2 \quad \text{w/ } (♠\text{-2}).$$

- scaling $\boldsymbol{v} := \boldsymbol{y}/\rho$ (as usually done in ADMM):

$$L_\rho(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{v}) = f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{v}\|^2 - \frac{\rho}{2}\|\boldsymbol{v}\|^2 \quad \text{w/ } (♠\text{-2}).$$

our method (2/3)

- We now have augmented Lagrangian:

$$L_\rho(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{v}) = f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{v}\|^2 - \frac{\rho}{2}\|\boldsymbol{v}\|^2$$

with $\boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{0}$, "$\boldsymbol{z}$ from a list".

- ADMM iteration:

$\boldsymbol{x}^{k+1} \leftarrow$ minimizer of $L_\rho(\boldsymbol{x}, \boldsymbol{z}^k; \boldsymbol{v}^k)$, (primal update-1)

$\boldsymbol{z}^{k+1} \leftarrow$ minimizer of $L_\rho(\boldsymbol{x}^{k+1}, \boldsymbol{z}; \boldsymbol{v}^k)$, (primal update-2)

$\boldsymbol{v}^{k+1} \leftarrow \boldsymbol{v}^k + \boldsymbol{x}^{k+1} - \boldsymbol{z}^{k+1}$. (dual update)

our method (2/3)

- We now have augmented Lagrangian:

$$L_\rho(\boldsymbol{x}, z; \boldsymbol{v}) = f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{x} - z + \boldsymbol{v}\|^2 - \frac{\rho}{2}\|\boldsymbol{v}\|^2$$

with $\boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{0}$, "$z$ from a list".

- primal update:

$$\boldsymbol{x}^{k+1} \leftarrow \text{minimizer of } L_\rho(\boldsymbol{x}, \boldsymbol{z}^k; \boldsymbol{v}^k), \qquad \text{(primal update-1)}$$

$$z^{k+1} \leftarrow \text{minimizer of } L_\rho(\boldsymbol{x}^{k+1}, z; \boldsymbol{v}^k), \qquad \text{(primal update-2)}$$

- update of $\boldsymbol{x}$:

$$\begin{aligned} \text{Min.} \quad & f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{x} - z^k + \boldsymbol{v}^k\|^2 \\ \text{s. t.} \quad & \boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{0}. \end{aligned}$$

  - apply nonlin. prog. approach
    - only local optimality (but, sufficient for heuristic)

our method (2/3)

- We now have augmented Lagrangian:

$$L_\rho(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{v}) = f(\boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{v}\|^2 - \frac{\rho}{2}\|\boldsymbol{v}\|^2$$

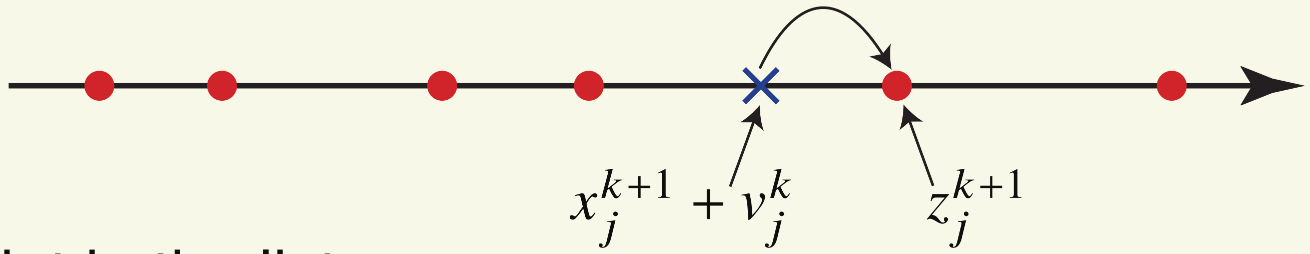
with $\boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{0}$, "$\boldsymbol{z}$ from a list".

- update of $\boldsymbol{z}$:

$$\text{Min.} \quad \frac{\rho}{2}\|(\boldsymbol{x}^{k+1} + \boldsymbol{v}^k) - \boldsymbol{z}\|^2$$

s. t. "$\boldsymbol{z}$ from a list".

  - $\equiv$ rounding $\boldsymbol{x}^{k+1} + \boldsymbol{v}^k$ to the nearest point in the list



  - ● : point in the list

our method (3/3)

- heuristic for nonconvex problems:
  - not guaranteed to converge
  - solution may depend on initial point & penalty parameter
    - not guaranteed to be globally optimal
- but, often effective

numerical experiments

- Matlab implementation
  - NLP solver: `fmincon` (Matlab built-in func.)
- multiple restart of ADMMM
  - from 100 random initial points
- penalty parameter $\rho = 100$
  - $\rho \leftarrow 10\rho$ if ADMM converges to an infeasible solution.
- 3 benchmark problems

ex.) 3-bar truss

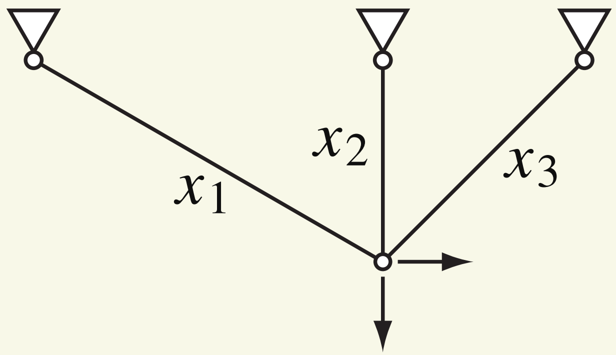
- truss w/ discrete member cross-sections [Rao '96]

$$
\begin{aligned}
\text{Min.} \quad & 2x_1 + x_2 + \sqrt{2}x_3 \\
\text{s. t.} \quad & \frac{\sqrt{3}x_2 + 1.932x_3}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_1x_3} \le 1, \\
& \frac{0.634x_1 + 2.828x_3}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_1x_3} \le 1, \\
& -1 \le \frac{0.5x_1 - 2x_2}{1.5x_1x_2 + \sqrt{2}x_2x_3 + 1.319x_1x_3} \le 1, \\
& x_1, x_2, x_3 \in \{0.1, 0.2, 0.3, 0.5, 0.8, 1.0, 1.2\}. \qquad \text{(disc.)}
\end{aligned}
$$

- 3 disc. variables
- 4 nonlin. ineq. cstr.

ex.) 3-bar truss

- truss w/ discrete member cross-sections [Rao '96]
  - $x_1, x_2, x_3 \in \{0.1, 0.2, 0.3, 0.5, 0.8, 1.0, 1.2\}$
  - min. structural weight under stress cstr.

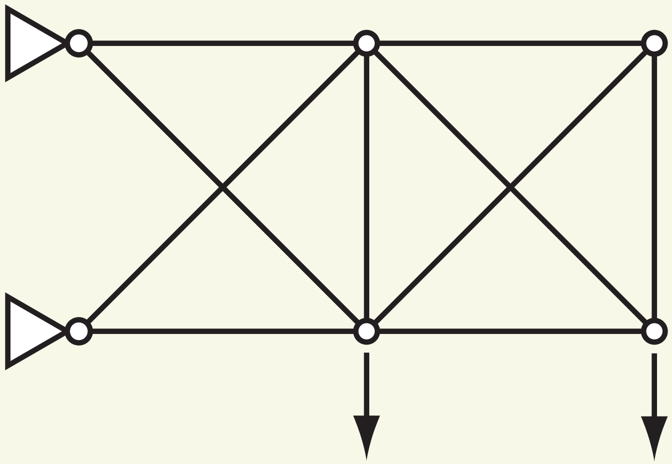


- proposed method finds global opt. $\boldsymbol{x}^* = (1.2, 0.5, 0.1)$

#trial	#conv.	#best	#iter.	time	final ρ
100	36	36	4.0	0.1 s	10^2

ex.) 10-bar truss

- truss w/ discrete member cross-sections [Cai & Thierauf, '93]
  - min. structural weight under stress & displacement cstr.
  - Case D1:
    $x_i \in \{1.62, 1.80, 1.99, 2.13, \dots, 33.50\}$ (42 candidates)
  - Case D2:
    $x_i \in \{0.100, 0.347, 0.440, 0.539, \dots, 33.700\}$ (30 candidates)



ex.) 10-bar truss

- truss w/ discrete member cross-sections [Cai & Thierauf, '93]
- proposed method finds:
  - $obj = 5490.74$ (case D1) [best known]
  - $obj = 5113.47$ (case D2) [2nd-best known]

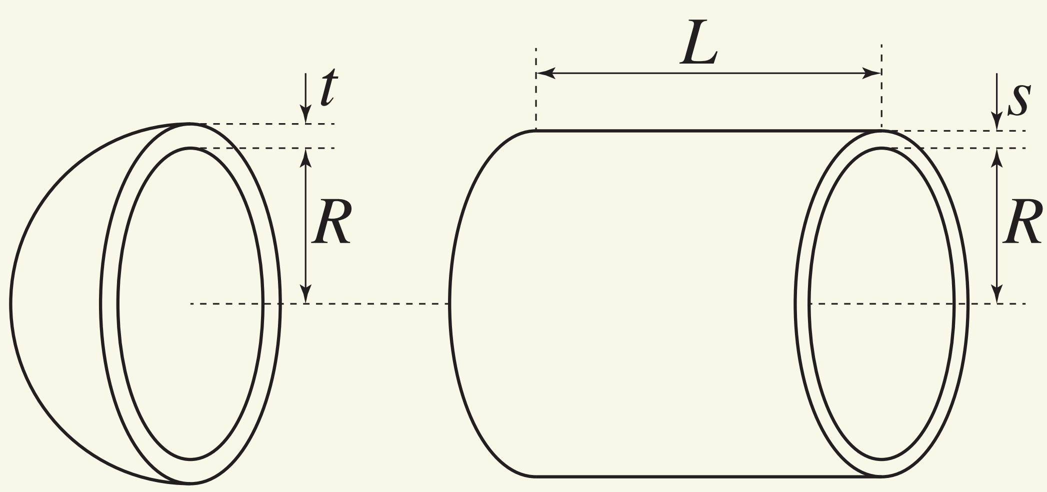
case	#trial	#conv.	#best	#iter.	time	final ρ
D1	100	80	4	18.6	1.4 s	10^3
D2	100	59	8	12.1	0.9 s	10^2

- existing results in literature:
  - case D1: $obj = 5490.74$
    [Groenwold, Stander, & Snyman, '99], [Juang & Chang, '06]
  - case D2: $obj = 5100.32$ [Thierauf & Cai, '98]

ex.) pressure vessel

- variables: 2 discrete & 2 continuous [Sandgren '90]

Min.	$0.6224 x_1 x_3 x_4 + 1.7781 x_2 {x_3}^2 + 3.1661 {x_1}^2 x_4 + 19.84 {x_1}^2 x_3$	(nonconvex)
s. t.	$\pi {x_3}^2 x_4 + \frac{4}{3}\pi {x_3}^3 \geq 1296000$	(nonconvex)
	$x_1 \geq 0.0193 x_3, \quad x_2 \geq 0.00954 x_3$	
	$10 \leq x_3 \leq 200, \quad 10 \leq x_4 \leq 200$	
	$x_1, x_2 \in \{1 \times 0.0625, 2 \times 0.0625, \dots, 99 \times 0.0625\}$	(disc.)



ex.) pressure vessel

- proposed method finds $obj = 6059.71$

#trial	#conv.	#best	#iter.	time	final ρ
100	100	100	10.3	0.4 s	10^3, 10^4

- existing results of metaheuristics:

	obj
[Kitayama, Arakawa, & Yamazaki ’06]	6029.87
[He, Prempain, & Wu, ’04]	6059.71
[Hsu, Sun, & Leu, ’95]	7021.67
[Kannan & Kramer, ’94]	7198.20
[Qian, Yu, & Zhou, ’93]	7238.83
[Sandgren, ’90]	8129.80

- → 2nd-best known solution was obtained.

conclusions

- heuristic for MINLP (mixed-integer nonlinear programming)
  - based on ADMM (alternating direction method of multipliers)
  - multiple restart from random initial points
- proposed ADMM
  - no guarantee, but...
  - simple, computationally cheap, and...
    - each iteration consists of an NLP and a projection
  - effective
    - i.e., often finds a feasible solution w/ good objective value