Improved Outer Approximation Methods for MINLP in Process System Engineering

Lijie Su, Lixin Tang, Ignacio E. Grossmann

The Logistics Institute, Northeastern University, China
Department of Chemical Engineering, Carnegie Mellon University

Outline

- MINLP is often applied to formulate complex problems for process industries, like synthesis, cracking scheduling, blending problems, etc.
- In order to solve MINLPs efficiently, we present three improved strategies for Outer Approximation(OA),
  - *Multiple-generation Cuts, Hybrid with GBD, Partial Surrogate Cuts*.
- Five improved OA algorithms are presented.
- Numerical experiments results illustrate the efficiency of those algorithms.

Introduction

- MINLP formulation

$$\min_{x,y} \quad z = f(x,y)$$
$$s.t. \quad g(\mathrm{x},\mathrm{y}) \le 0$$
$$Ax + Ey \le b$$
$$\mathrm{x} \in \mathrm{X},\ y \in Y$$

$$\min_{x,y} \quad z = f(x) + c^T y$$
$$s.t. \quad g(\mathrm{x}) + Cy \le 0$$
$$Ax + Ey \le b$$
$$\mathrm{x} \in \mathrm{X},\ y \in Y$$

- Assume that f and g are continuous differentiable and convex on the compact polyhedral convex set X, and Y is a finite discrete set.
- Usually there are no discrete variables in nonlinear terms.

Review

- MINLP algorithms
  - Outer Approximation, OA (Duran&Grossmann1986, Fletcher& Leyffer1994)
  - Generalized Benders Decomposition(GBD, Geoffrion1972),
  - Branch and Bound, B&B (Gupta and Ravindran, 1985)
  - Branch and cut (Stubbs&Mehrotra, 1999)
  - LP/NLP based B&B (Quesda&Grossmann1991),
  - Extended Cutting Plane, ECP (Westerlund&Pettersson, 1995)
  -
- OA and GBD are two common methods for solving MINLP.

Motivation

- Limitation of OA
  - MILP master may be computationally expensive
- Limitation of GBD
  - Weak lower bounds (slow convergence)
- Limitation of Both
  - Potentially many infeasible NLP subproblems
- Our work aims at designing improved OA algorithms to increase solution efficiency.

OA & GBD Algorithms

- OA / GBD algorithms
  - Decomposition methods with iterations between NLP subproblem and MILP master problem
  - OA: Master MILP

$$\min \quad \alpha$$

$$s.t. \quad \alpha \geq f\left(x^k, y^k\right) + \nabla f\left(x^k, y^k\right)\begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix}$$

$$g\left(x^k, y^k\right) + \nabla g\left(x^k, y^k\right)\begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} \leq 0$$

$$Ax + Ey \leq 0$$

$$x \in X$$

GBD: Master MILP

$$\min \quad \alpha$$

$$s.t. \quad \alpha \geq f\left(x^k, y^k\right) + \nabla_y f\left(x^k, y^k\right)\left(y - y^k\right)$$

$$+ \lambda^T\left(g\left(x^k, y^k\right) + \nabla_y g\left(x^k, y^k\right)\left(y - y^k\right)\right)$$

$$y \in Y$$

(x^k, y^k) : solution of NLP subproblem

  - MILP GBD: cuts are surrogate cuts of OA cutting planes

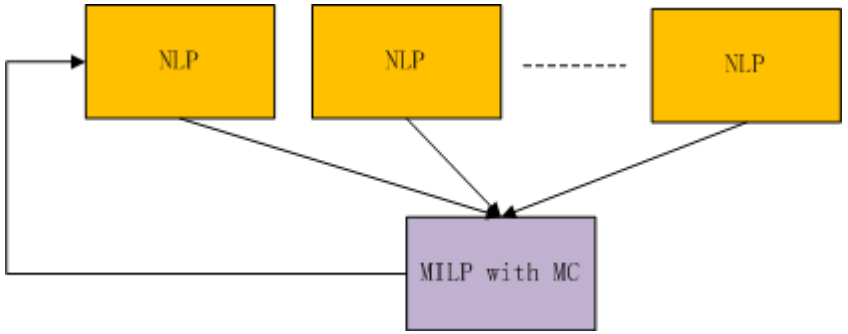
Three Improved OA Strategies

- Three improved strategies are presented.
  - Multi-generation Cuts
  - Hybrid with GBD
  - Partial Surrogate Cuts

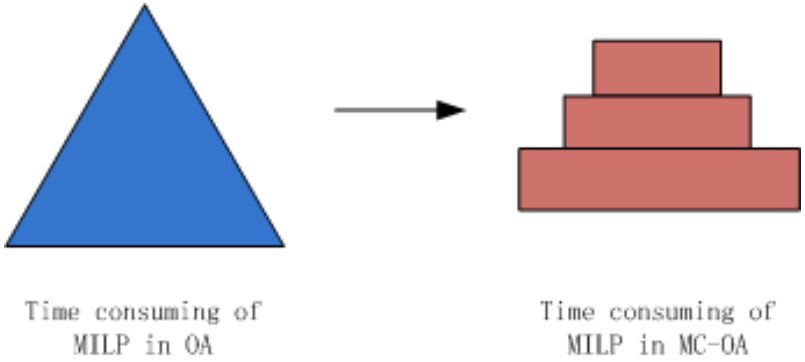
Multi-generation Cuts(MC)

- Idea : Increase the number of cuts generated at each iteration to improve the approximation of the MILP.

  Multiple NLPs are solved at each iteration.



- Goal:
  - Accelerate convergence
  - Reduce the number of iterations.



Multi-generation Cuts(MC)

- S is the set of cuts generated in one iteration
  - OA

$$\eta \geq f\left(x^{k,s}, y^{k,s}\right) + \nabla f\left(x^{k,s}, y^{k+1,s}\right)\begin{pmatrix} x - x^{k,s} \\ y - y^{k,s} \end{pmatrix}$$

$$g\left(x^{k,s}, y^{k,s}\right) + \nabla g\left(x^{k,s}, y^{k,s}\right)\begin{pmatrix} x - x^{k,s} \\ y - y^{k,s} \end{pmatrix} \leq 0 \quad s \in S$$

  - GBD

$$\alpha \geq f\left(x^{k,s}, y^{k,s}\right) + \nabla_y f\left(x^{k,s}, y^{k,s}\right)\left(y - y^{k,s}\right)$$
$$+ \lambda^T\left(g\left(x^{k,s}, y^{k,s}\right) + \nabla_y g\left(x^{k,s}, y^{k,s}\right)\left(y - y^{k,s}\right)\right) \quad s \in S$$

- **Theorem 1**. The MC strategy does not change the optimal solution of OA or GBD algorithm for convex MINLP problems.

Multi-generation Cuts(MC)

- Algorithm for set of cuts- S
  - Step1. Solve MILP master problem to obtain (S):
    - Obtain suboptimal integer solutions
    - Obtain optimal integer solution
  - Step2. Solve multiple NLP subproblems from Step 1 with fixed integer variables.
    - Can be solved in parallel
  - Step3. Obtain cutting planes/cuts from NLP subproblem in Step 2 and add to MILP master problem. Go to Step1.

Multi-generation Cuts(MC)

- Remark
  - The size of set S influences the efficiency of MC.
  - Large size improves lower bound, increases computation
  - Small size leads to weaker bounds but less computation

Hybrid with GBD

- Based on simpler master problem for GBD and tighter lower bound for OA. Strategy of hybrid with GBD decomposes the solution procedure of the MINLP into two stages.
  - Stage1. GBD is used initially to construct a rough approximation of projected feasible region.
  - Stage2. OA is used in the subsequent iterations in order to enforce the convergence of solution procedure.

Hybrid with GBD

- Set the total iteration M for GBD cuts in first stage

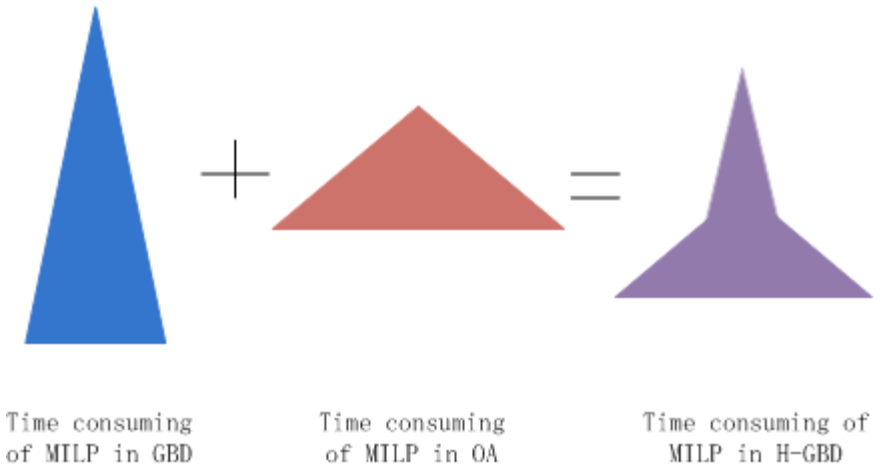
$$\eta \geq f\left(x^k, y^k\right) + \nabla f_y\left(x^k, y^k\right)\left(y - y^k\right) +$$
$$\left(\lambda^k\right)^T \cdot \left(g\left(x^k, y^k\right) + \nabla g_y\left(x^k, y^k\right)\left(y - y^k\right)\right) \quad k = 1, \ldots, M$$

- Switch to OA cuts from iteration M +1 until termination

$$\eta \geq f\left(x^k, y^k\right) + \nabla f\left(x^k, y^k\right)\begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix}$$
$$g\left(x^{k,s}, y^k\right) + \nabla g\left(x^k, y^k\right)\begin{pmatrix} x - x^k \\ y - y^k \end{pmatrix} \leq 0 \quad k = M + 1, \ldots$$

Hybrid with GBD

- Remark 1. The bounds from the first stage are inherited into the second stage. The initial master problem of OA in this improved procedure inherits all GBD generation cuts from stage 1.
- Remark 2. The CPU times are affected by the setting limitations on GBD iterations, and also dependent on the problems.



Hybrid with GBD

- **Theorem 2.** The hybrid strategy of OA and GBD does not change the optimal solution of OA or GBD algorithm for convex MINLP problems.
- Discussion on the case of infeasible NLP subproblem.
  - The infeasible cuts generating from GBD are weak, especially the case of nonlinear terms or constraints.
  - OA infeasible cuts are stronger than GBD cuts when there are nonlinear constraints.
  - Extension: When the NLP subproblem is infeasible, obtain OA infeasible cuts instead GBD cuts in GBD stage. Add GBD generation cuts when the NLP is feasible.

Partial Surrogate Cuts

- PSC (Quesada&Grossmann,1991)
  - MINLP models with many linear constraints

MINLP formulation

$$Z = \min \quad c^T y + a^T w + f(v)$$
$$s.t. \quad Cy + Dw + g(v) \le 0$$
$$Ey + A_1 w + A_2 v \le b$$
$$y \in Y, \begin{pmatrix} w \\ v \end{pmatrix} = x, \; x \in X$$

y and *w* linear variables

v nonlinear variables

PSC

(projects only onto nonlinear terms)

$$\alpha \ge c^T y + a^T w + f(v^k) + (\lambda^k)^T \left[Cy + Dw + g(v^k) \right] - (\mu^k)^T \mathrm{A}_2 (\mathrm{v} - v^k)$$
$$Ey + \mathrm{A}_1 w + \mathrm{A}_2 v \le b$$

v^k is the solution of NLP

Partial Surrogate Cuts

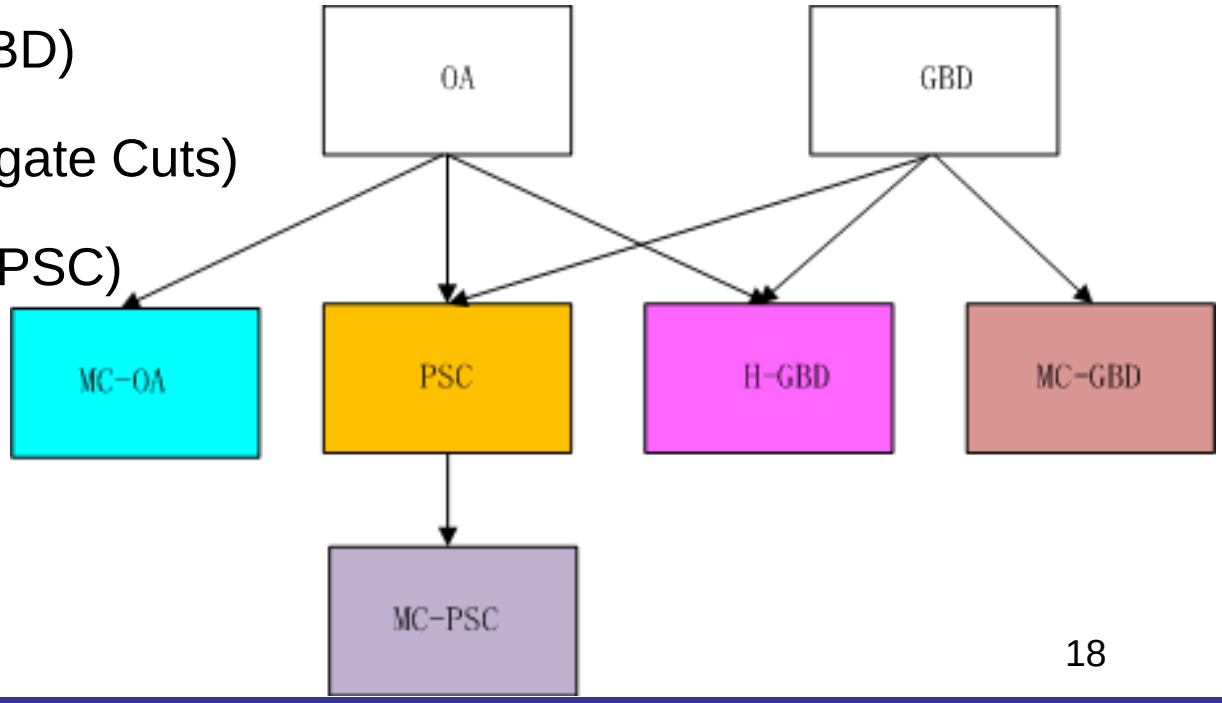
- Advantage of PSC: Cuts yield tighter LB than GBD cut, and involves fewer constraints than OA cuts.

$$LB_{GBD} \leq LB_{PSC} \leq LB_{OA}$$

- Idea: using PSC instead of OA cuts in OA algorithm.
- **Theorem 3.** The decomposition method based on PSC converges to the optimal solution of convex MINLP.

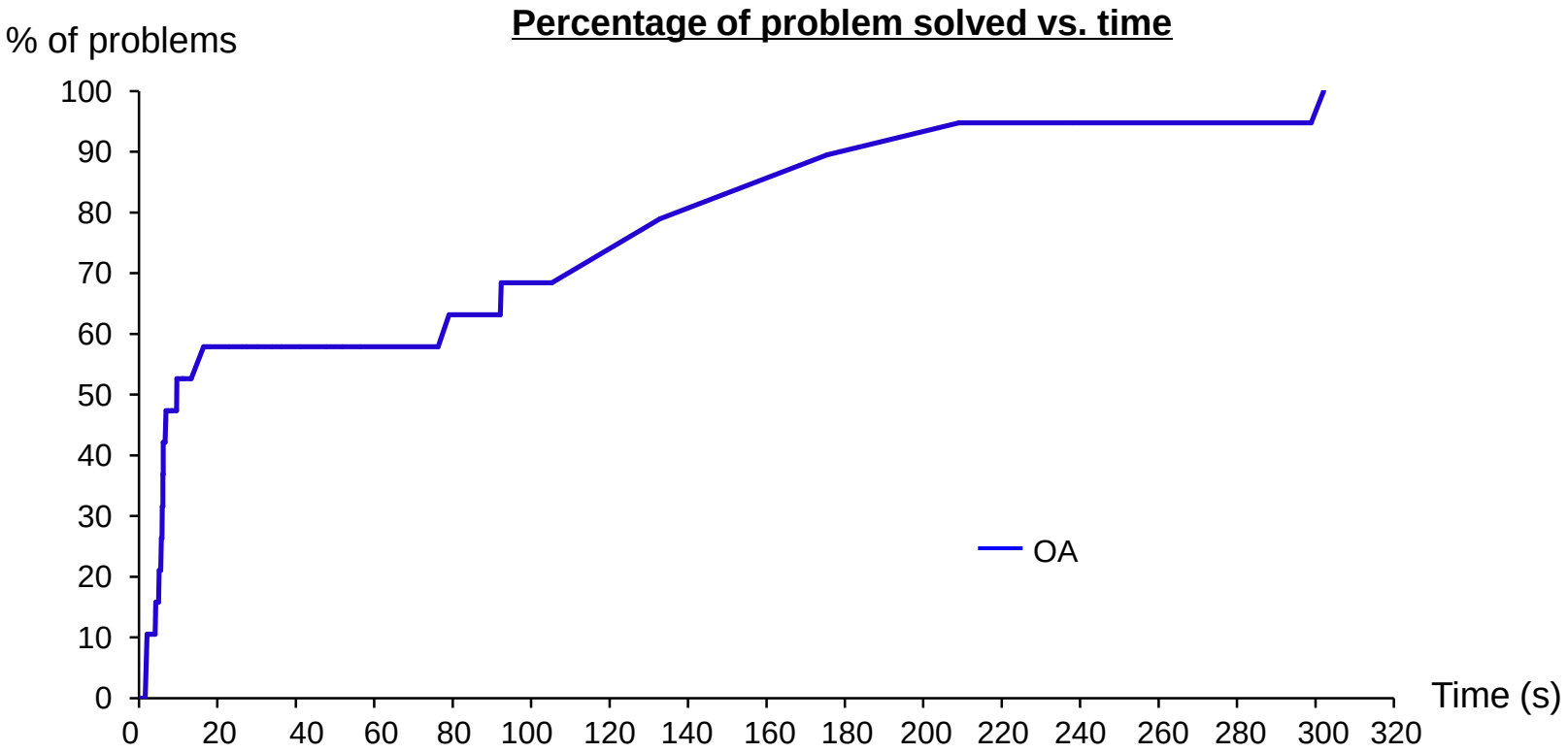
Improved OA algorithms

- We present five improved OA algorithms based on the proposed strategies.
  - MC-OA (Mutlicut OA)
  - MC-GBD (Mutlicut GBD)
  - H-GBD (Hybrid GBD)
  - PSC (Partial Surrogate Cuts)
  - MC-PSC (Multicut PSC)



MC-OA compared to OA

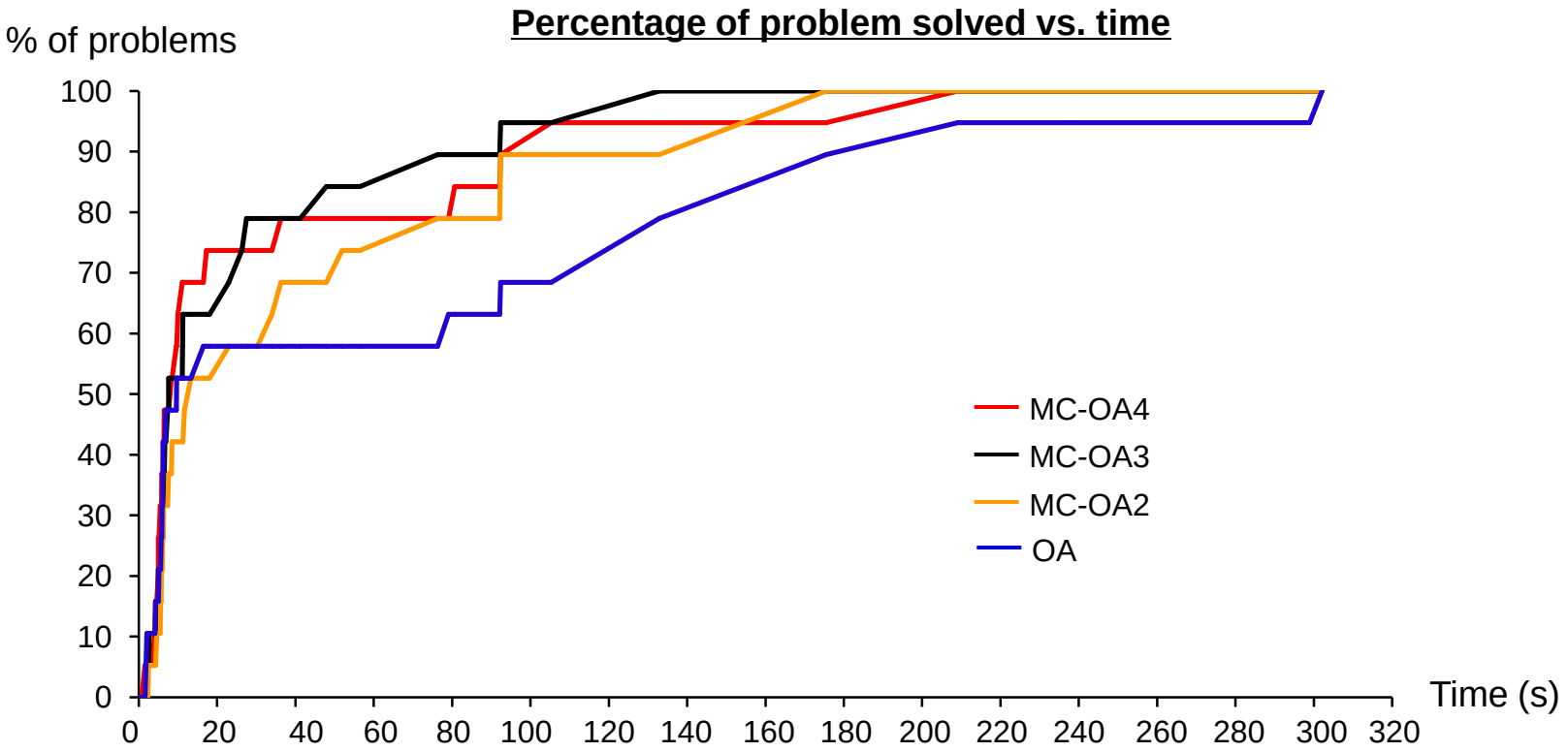
19 test instances（minlp.org, minlplib）



Note: MINLP problems solved with CONOPT/CPLEX, in GAMS 24.2.2 in a 2.6 GHz Processor, Intel® Core™ i7. 7GB of RAM.

MC-OA solves problems faster than OA

19 test instances（minlp.org, minlplib）

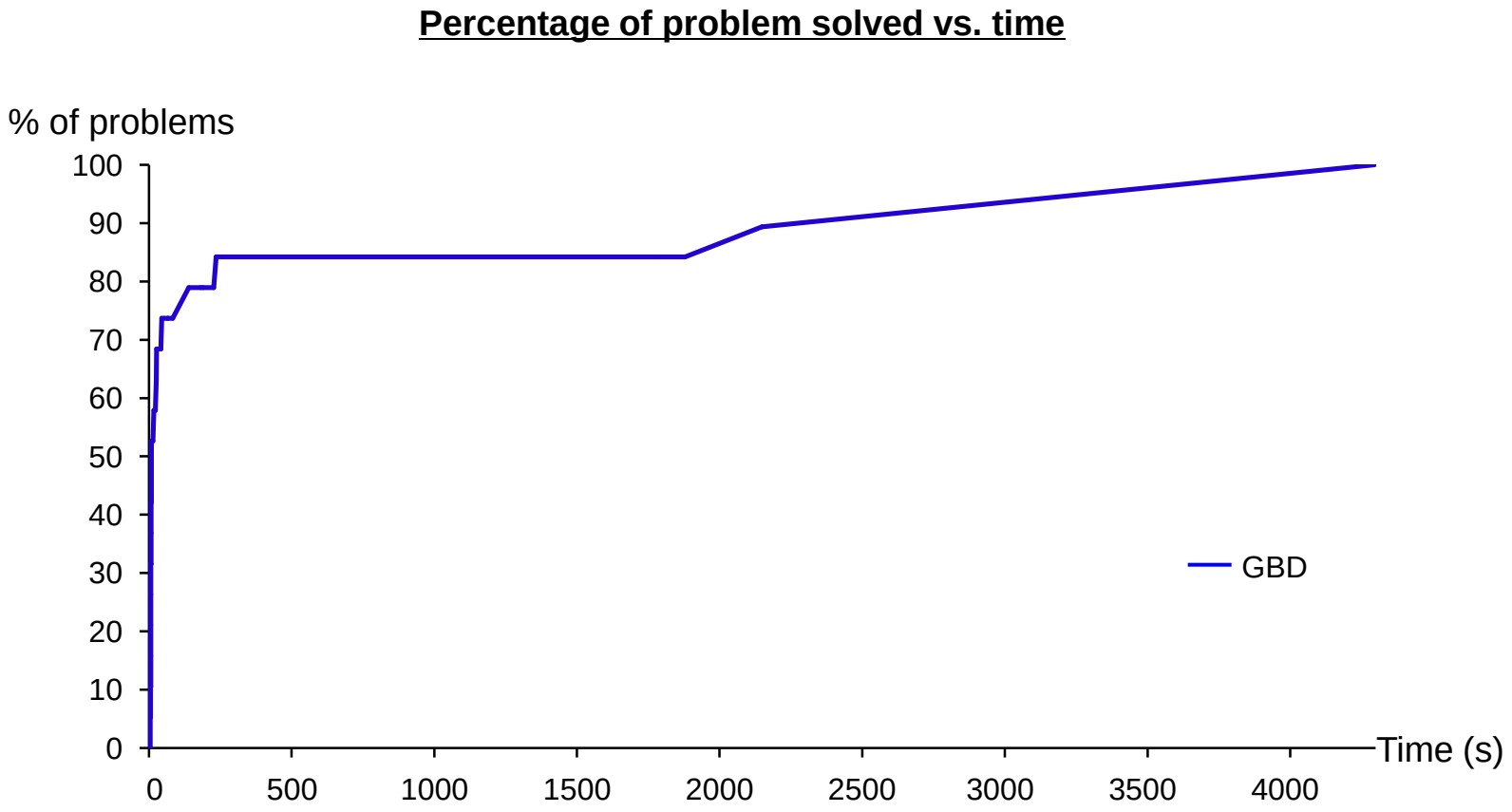


- |S| are set 2, 3, and 4.
- With the increase of problem size, the improvements of MC-OAs are significant.

Note: MINLP problems solved with CONOPT/CPLEX, in GAMS 24.2.2 in a 2.6 GHz Processor, Intel® Core™ i7. 7GB of RAM.

MC-GBD compared to GBD

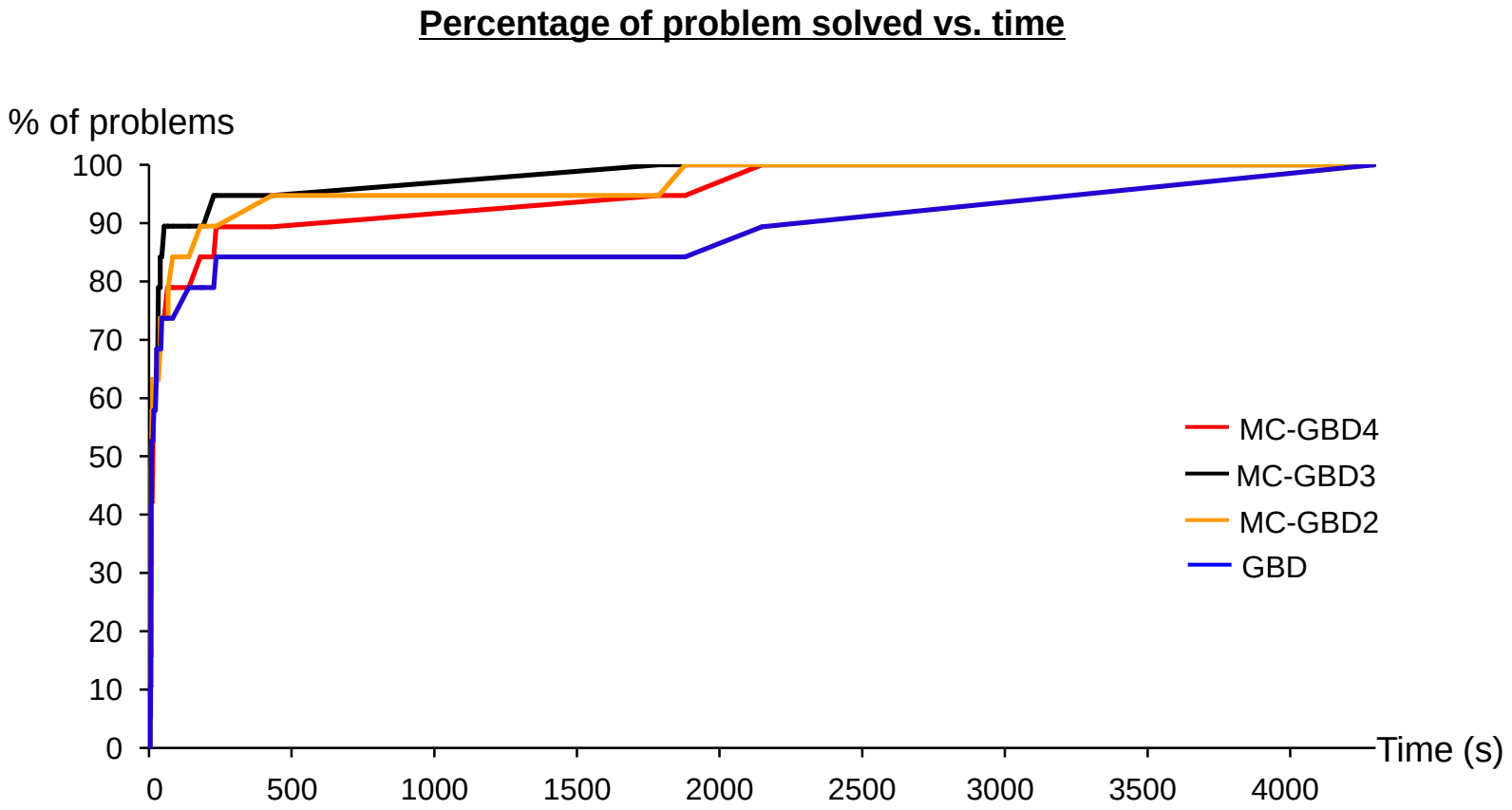
19 test instances



Note: MINLP problems solved with SBB/CONOPT, in GAMS 24.1.3 in a 2.93 GHz Processor, Intel® Core™ i7. 4GB of RAM.

MC-GBD solves faster than GBD

19 test instances

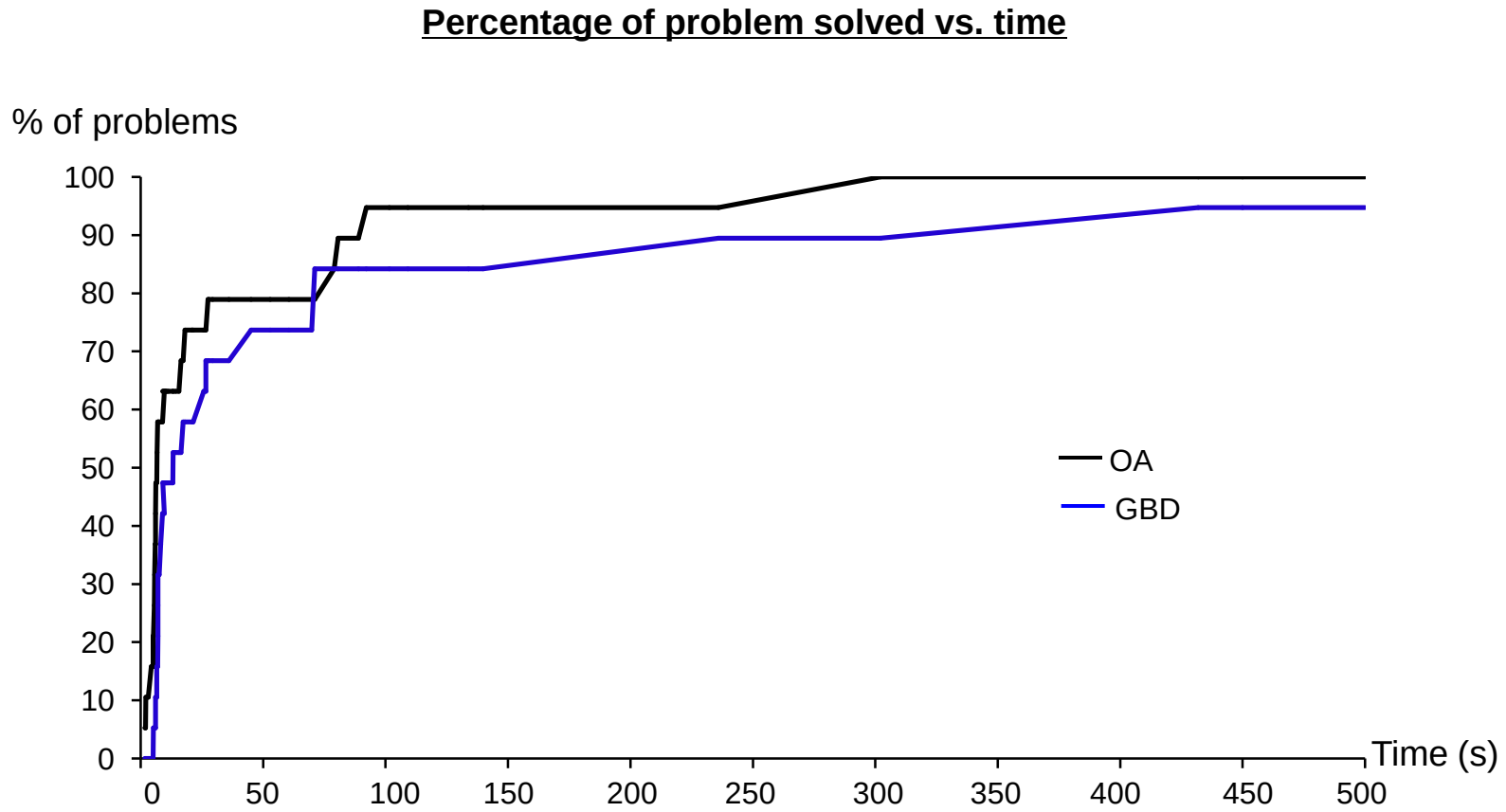


- The performance of MC-GBDs are not as good as MC-OAs.

Note: MINLP problems solved with SBB/CONOPT, in GAMS 24.1.3 in a 2.93 GHz Processor, Intel® Core™ i7. 4GB of RAM.

H-GBD compared to OA and GBD

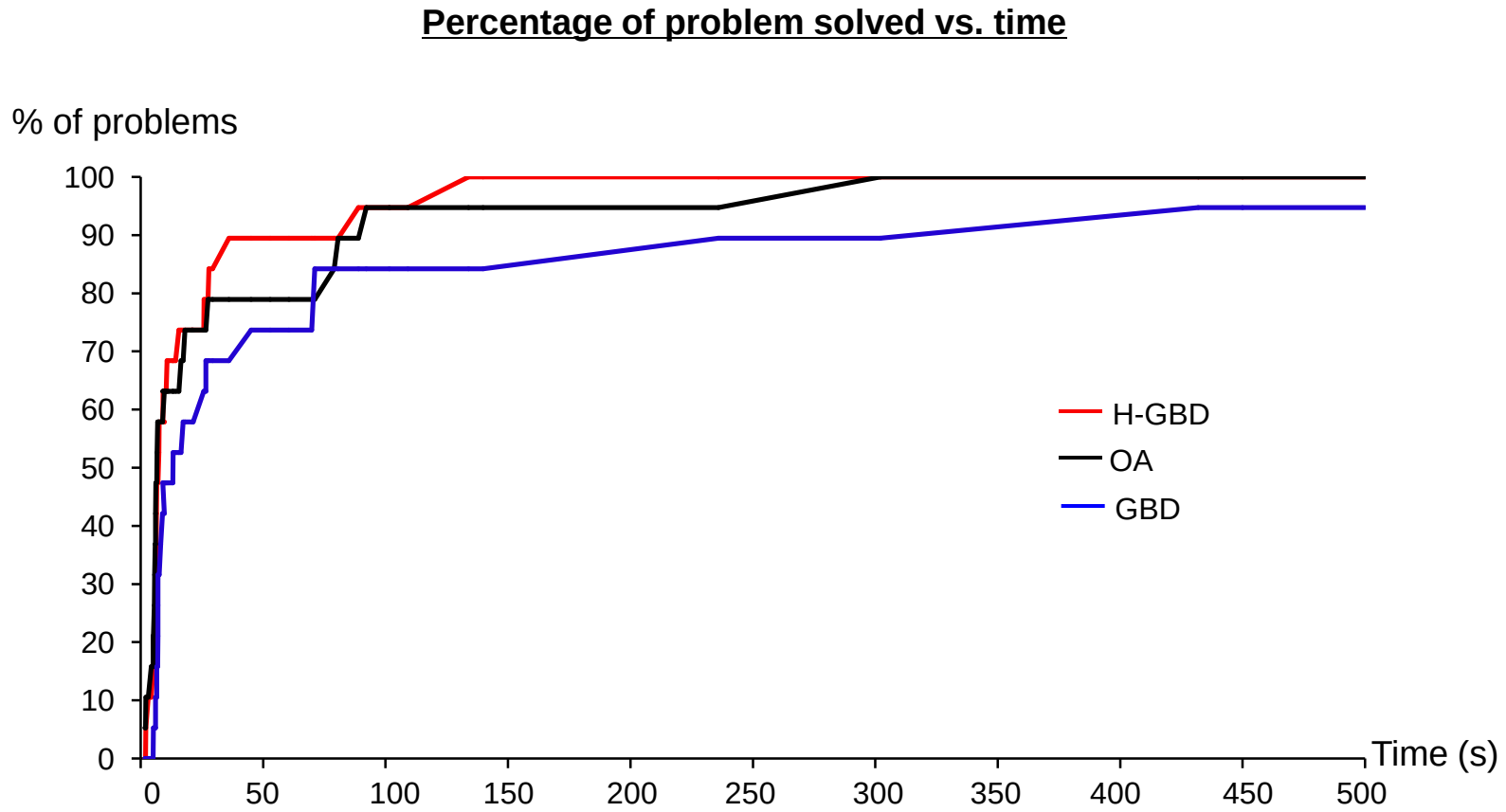
19 test instances



Note: MINLP problems solved with CONOPT/CPLEX, in GAMS 24.2.2 in a 2.6 GHz Processor, Intel® Core™ i7. 7GB of RAM.

H-GBD solves faster than OA in some instances

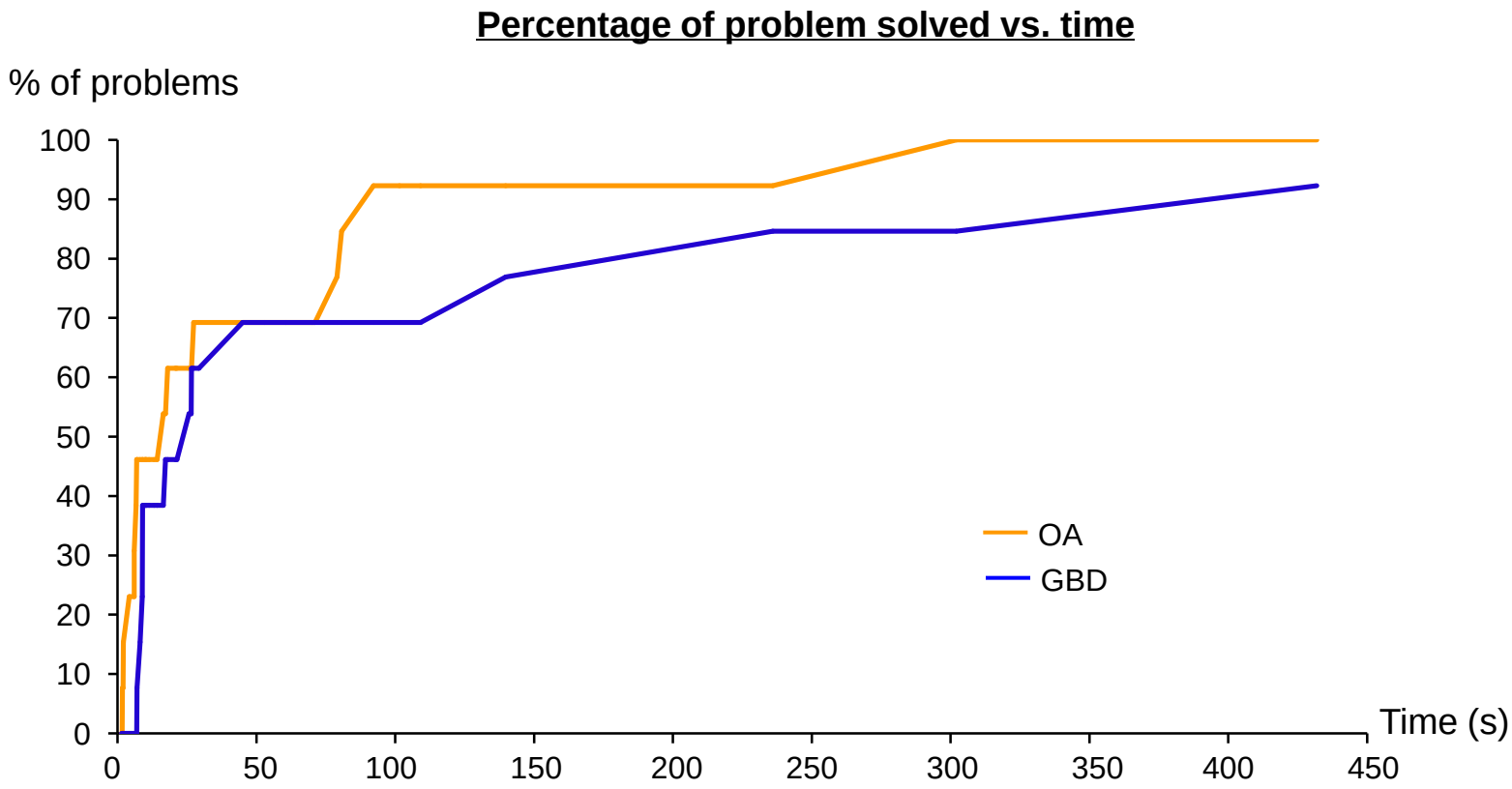
19 test instances



Note: MINLP problems solved with CONOPT/CPLEX, in GAMS 24.2.2 in a 2.6 GHz Processor, Intel® Core™ i7. 7GB of RAM.

PSC, MC-PSC2 compared to OA and GBD

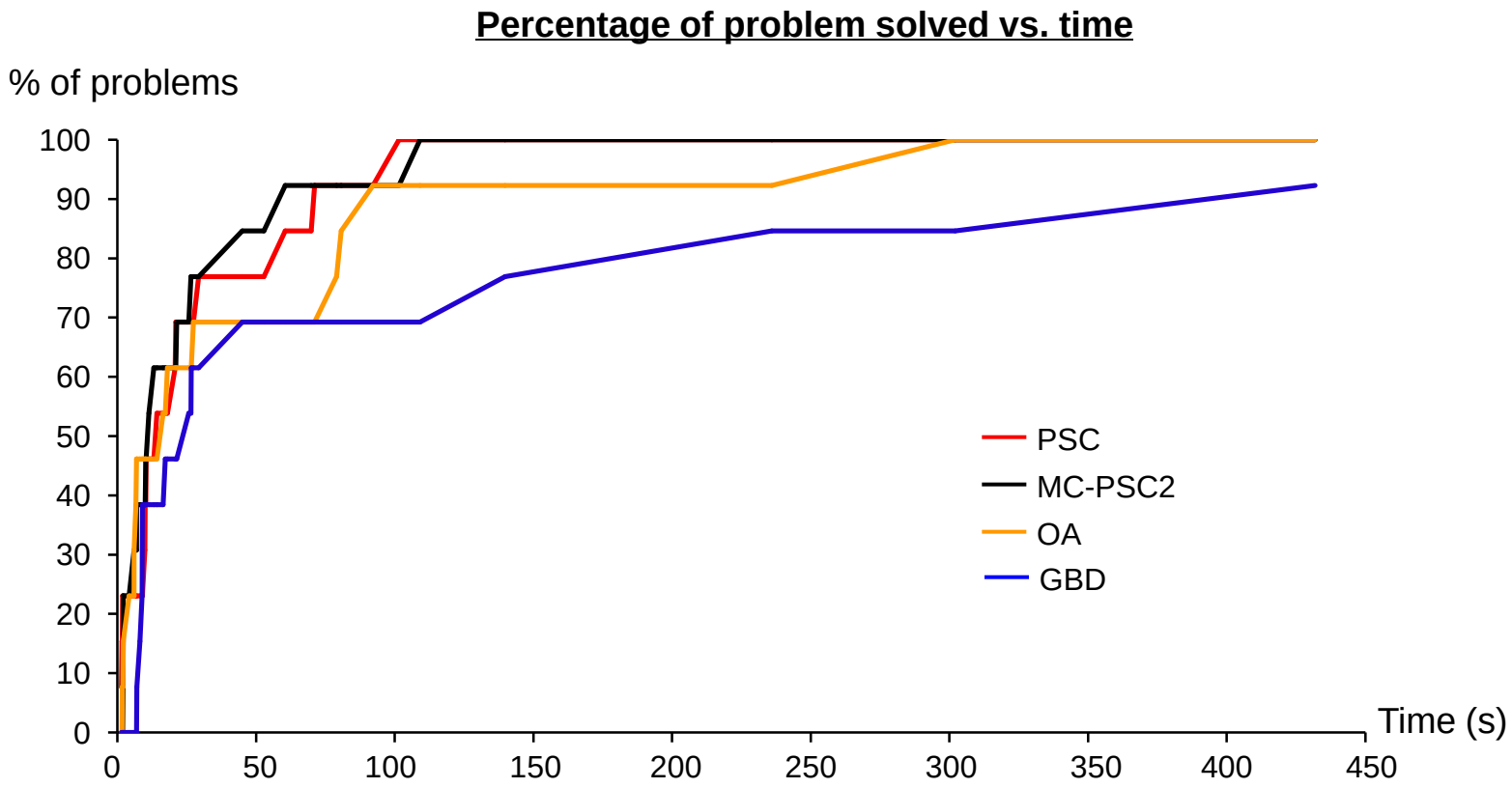
13 test instances(subset of larger problems)



Note: MINLP problems solved with CONOPT/CPLEX, in GAMS 24.2.2 in a 2.6 GHz Processor, Intel® Core™ i7. 7GB of RAM.

PSC, MC-PSC2 improved performance compared to OA

13 test instances(subset of larger problems)



Note: MINLP problems solved with CONOPT/CPLEX, in GAMS 24.2.2 in a 2.6 GHz Processor, Intel® Core™ i7. 7GB of RAM.

Conclusions

- Five improved OA/GBD algorithms are studied to increase solution efficiency, which are MC-OA, MC-GBD, H-GBD, PSC and MC-PSC.
- Computational results illustrate the efficiency of those methods.
  - MC-OAs perform best on average among all methods.
  - H-GBD can be faster than OA on some instances.
  - PSC improved performance compared to OA.

Thank you for your attention
Any questions?

Lijie Su, Lixin Tang, Ignacio E. Grossmann

The Logistics Institute, Northeastern University
Department of Chemical Engineering, Carnegie Mellon University