A SHOT AT CONVEX OPTIMIZATION

17th British-French-German Conference on Optimization
15–17 June 2015, London, United Kingdom

Andreas Lundell, Jan Kronqvist and Tapio Westerlund

Optimization and Systems Engineering
Åbo Akademi University, Finland

ose OPTIMIZATION AND SYSTEMS ENGINEERING

Åbo Akademi University

CONTENTS OF THE TALK

The extended supporting hyperplane (ESH) algorithm is a method for solving convex MINLP problems to global optimality by solving sequences of LP and MILP problems.

CONTENTS OF THE TALK

The extended supporting hyperplane (ESH) algorithm is a method for solving convex MINLP problems to global optimality by solving sequences of LP and MILP problems.

The supporting hyperplane optimization toolkit (SHOT) is a new solver for convex MINLP:

- Incorporates the ESH algorithm and primal heuristics.
- To be released as an open source COIN-OR project

CONTENTS OF THE TALK

The extended supporting hyperplane (ESH) algorithm is a method for solving convex MINLP problems to global optimality by solving sequences of LP and MILP problems.

The supporting hyperplane optimization toolkit (SHOT) is a new solver for convex MINLP:

- Incorporates the ESH algorithm and primal heuristics.
- To be released as an open source COIN-OR project

Results from an **extensive benchmark of SHOT** against several other MINLP solvers is provided.

THE EXTENDED SUPPORTING HYPERPLANE ALGORITHM

THE EXTENDED SUPPORTING HYPERPLANE ALGORITHM

A new method for global optimization of convex MINLP problems.

THE EXTENDED SUPPORTING HYPERPLANE ALGORITHM

A new method for global optimization of convex MINLP problems.

Supporting hyperplanes describe the nonlinear feasible set:

- utilizes a line search procedure to find the generation point
- an interior point is required for the line search

THE EXTENDED SUPPORTING HYPERPLANE ALGORITHM

A new method for global optimization of convex MINLP problems.

Supporting hyperplanes describe the nonlinear feasible set:

- utilizes a line search procedure to find the generation point
- an interior point is required for the line search

Similar ideas as presented in:

> *Veinott Jr. A. F., The supporting hyperplane method for unimodal programming, Operations Research, Vol. 15, pp. 147–152, 1967*

THE EXTENDED SUPPORTING HYPERPLANE ALGORITHM

A new method for global optimization of convex MINLP problems.

Supporting hyperplanes describe the nonlinear feasible set:

- » utilizes a line search procedure to find the generation point
- » an interior point is required for the line search

Similar ideas as presented in:

> *Veinott Jr. A. F., The supporting hyperplane method for unimodal programming, Operations Research, Vol. 15, pp. 147–152, 1967*

The ESH algorithm and the SHOT solver is described in:

> *Kronqvist J., Lundell A. and Westerlund T., The extended supporting hyperplane algorithm for convex MINLP problems, Journal of Global Optimization, accepted 2015*

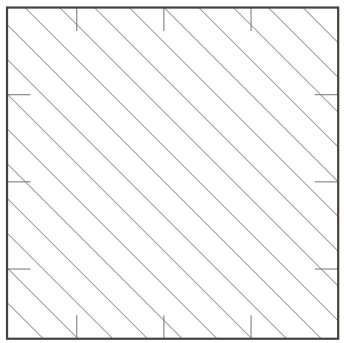
THE MINLP PROBLEM SCOPE

The ESH algorithm solves convex MINLP problems of the type

$$\text{find } x^* \in \arg\min_{x \in C \cap L \cap Y} c^T x \qquad \text{(P)}$$

where $x = [x_1, x_2, \ldots, x_N]^T$ belongs to the compact set

$$X = \{x \mid \underline{x}_i \leq x_i \leq \overline{x}_i,\ i = 1, \ldots, N\} \subset \mathbb{R}^n,$$



the feasible region is defined by $C \cap L \cap Y$

$$\begin{aligned} C &= \{x \mid g_m(x) \leq 0,\ m = 1, \ldots, M,\ x \in X\} \\ L &= \{x \mid Ax \leq a,\ Bx = b,\ x \in X\} \\ Y &= \{x \mid x_i \in \mathbb{Z},\ i \in I_{\mathbb{Z}},\ x \in X\} \end{aligned}$$

and C is a convex set.

THE MINLP PROBLEM SCOPE

The ESH algorithm solves convex MINLP problems of the type

$$\text{find } x^* \in \arg\min_{x \in C \cap L \cap Y} c^T x \qquad \text{(P)}$$

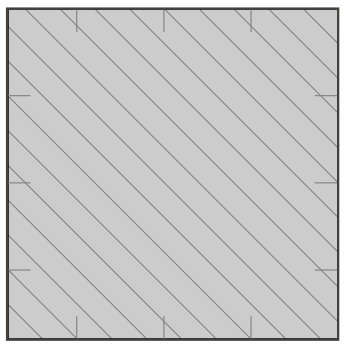
where $x = [x_1, x_2, \ldots, x_N]^T$ belongs to the compact set

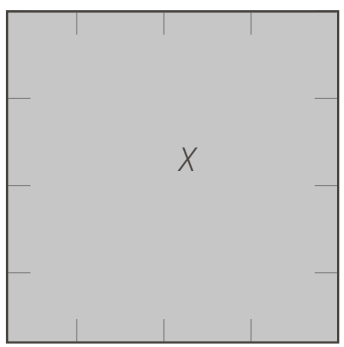
$$X = \{x \mid \underline{x}_i \leq x_i \leq \bar{x}_i,\ i = 1, \ldots, N\} \subset \mathbb{R}^n,$$

the feasible region is defined by $C \cap L \cap Y$

$$\begin{aligned} C &= \{x \mid g_m(x) \leq 0,\ m = 1, \ldots, M,\ x \in X\} \\ L &= \{x \mid Ax \leq a,\ Bx = b,\ x \in X\} \\ Y &= \{x \mid x_i \in \mathbb{Z},\ i \in I_{\mathbb{Z}},\ x \in X\} \end{aligned}$$

and C is a convex set.





THE MINLP PROBLEM SCOPE

The ESH algorithm solves convex MINLP problems of the type

$$\text{find } x^* \in \arg\min_{x \in C \cap L \cap Y} c^T x \qquad \text{(P)}$$

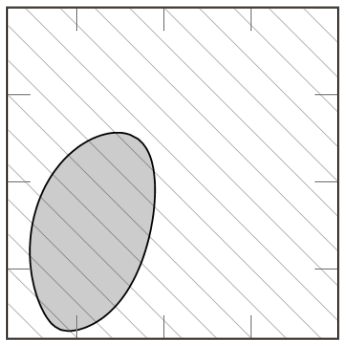
where $x = [x_1, x_2, \ldots, x_N]^T$ belongs to the compact set

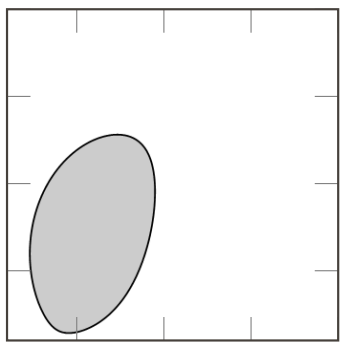
$$X = \{x \mid \underline{x}_i \leq x_i \leq \overline{x}_i,\ i = 1, \ldots, N\} \subset \mathbb{R}^n,$$

the feasible region is defined by $C \cap L \cap Y$

$$\begin{aligned} C &= \{x \mid g_m(x) \leq 0,\ m = 1, \ldots, M,\ x \in X\} \\ L &= \{x \mid Ax \leq a,\ Bx = b,\ x \in X\} \\ Y &= \{x \mid x_i \in \mathbb{Z},\ i \in I_{\mathbb{Z}},\ x \in X\} \end{aligned}$$

and C is a convex set.





THE MINLP PROBLEM SCOPE

The ESH algorithm solves convex MINLP problems of the type

$$\text{find } x^* \in \arg\min_{x \in C \cap L \cap Y} c^T x \qquad \text{(P)}$$

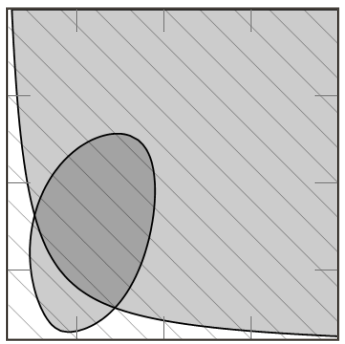
where $x = [x_1, x_2, \ldots, x_N]^T$ belongs to the compact set

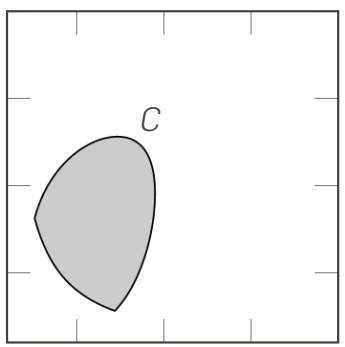
$$X = \{x \mid \underline{x}_i \leq x_i \leq \overline{x}_i,\ i = 1, \ldots, N\} \subset \mathbb{R}^n,$$

the feasible region is defined by $C \cap L \cap Y$

$$\begin{aligned} C &= \{x \mid g_m(x) \leq 0,\ m = 1, \ldots, M,\ x \in X\} \\ L &= \{x \mid Ax \leq a,\ Bx = b,\ x \in X\} \\ Y &= \{x \mid x_i \in \mathbb{Z},\ i \in I_{\mathbb{Z}},\ x \in X\} \end{aligned}$$

and C is a convex set.





THE MINLP PROBLEM SCOPE

The ESH algorithm solves convex MINLP problems of the type

$$\text{find } x^* \in \arg\min_{x \in C \cap L \cap Y} c^T x \qquad \text{(P)}$$

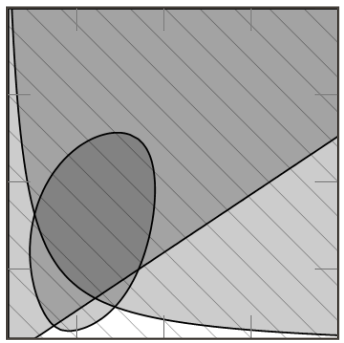
where $x = [x_1, x_2, \ldots, x_N]^T$ belongs to the compact set

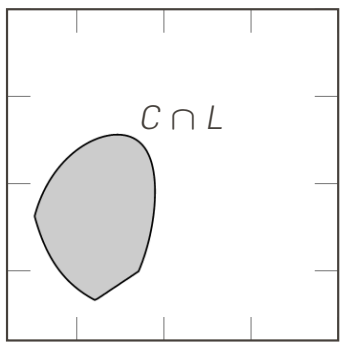
$$X = \{x \mid \underline{x}_i \leq x_i \leq \overline{x}_i,\ i = 1, \ldots, N\} \subset \mathbb{R}^n,$$

the feasible region is defined by $C \cap L \cap Y$

$$\begin{aligned} C &= \{x \mid g_m(x) \leq 0,\ m = 1, \ldots, M,\ x \in X\} \\ L &= \{x \mid Ax \leq a,\ Bx = b,\ x \in X\} \\ Y &= \{x \mid x_i \in \mathbb{Z},\ i \in I_{\mathbb{Z}},\ x \in X\} \end{aligned}$$

and C is a convex set.





THE MINLP PROBLEM SCOPE

The ESH algorithm solves convex MINLP problems of the type

$$\text{find } x^* \in \arg\min_{x \in C \cap L \cap Y} c^T x \qquad \text{(P)}$$

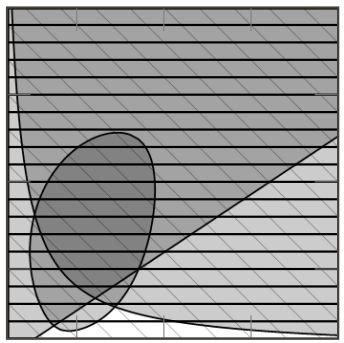
where $x = [x_1, x_2, \ldots, x_N]^T$ belongs to the compact set

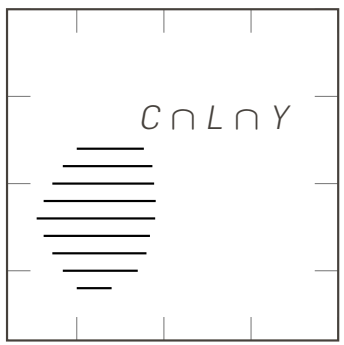
$$X = \{x \mid \underline{x}_i \leq x_i \leq \overline{x}_i,\ i = 1, \ldots, N\} \subset \mathbb{R}^n,$$

the feasible region is defined by $C \cap L \cap Y$

$$\begin{aligned} C &= \{x \mid g_m(x) \leq 0,\ m = 1, \ldots, M,\ x \in X\} \\ L &= \{x \mid Ax \leq a,\ Bx = b,\ x \in X\} \\ Y &= \{x \mid x_i \in \mathbb{Z},\ i \in I_{\mathbb{Z}},\ x \in X\} \end{aligned}$$

and C is a convex set.





BREAKDOWN OF THE ESH ALGORITHM

Interior point search step

Obtain a feasible, relaxed interior point (satisfying C) by solving a NLP problem.

LP step (optional)

Solve simple LP problems (initially in $X \cap L$) to obtain an initial overestimating linear set.

MILP step

Solve MILP problems to find the optimal solution to (P).

BREAKDOWN OF THE ESH ALGORITHM

Interior point search step

Obtain a feasible, relaxed interior point (satisfying C) by solving a NLP problem.

LP step (optional)

Solve simple LP problems (initially in $X \cap L$) to obtain an initial overestimating linear set.

MILP step

Solve MILP problems to find the optimal solution to (P).

BREAKDOWN OF THE ESH ALGORITHM

Interior point search step

Obtain a feasible, relaxed interior point (satisfying C) by solving a NLP problem.

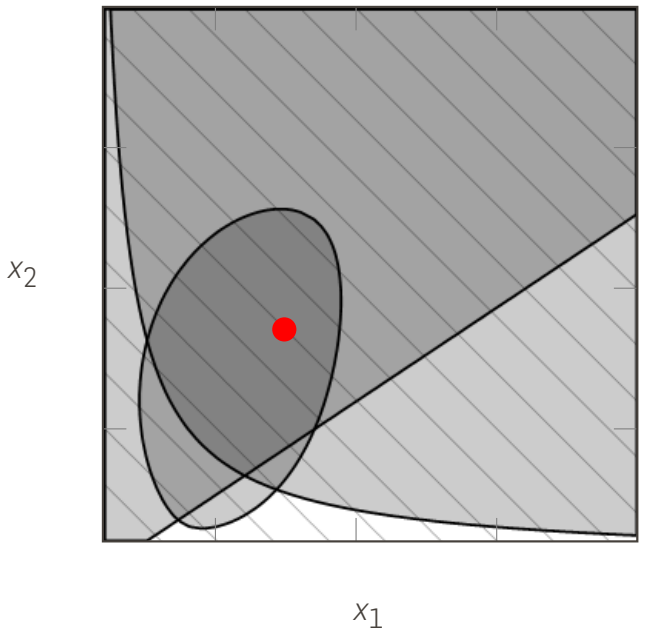
LP step (optional)

Solve simple LP problems (initially in $X \cap L$) to obtain an initial overestimating linear set.

MILP step

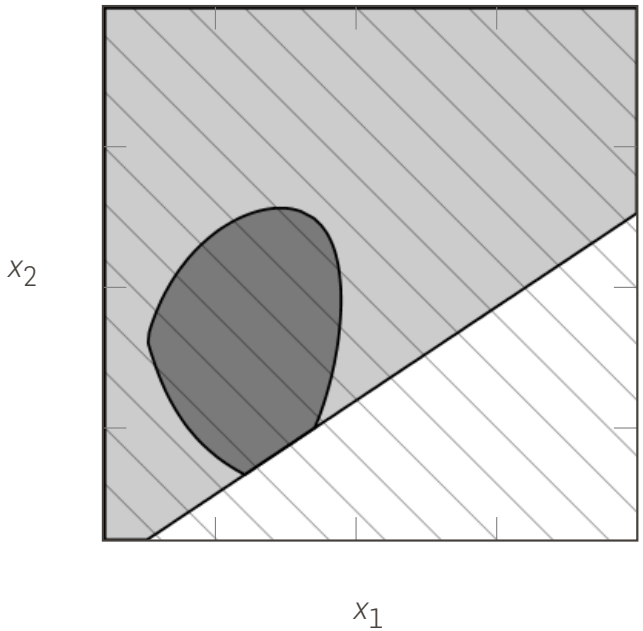
Solve MILP problems to find the optimal solution to (P).

INTERIOR POINT SEARCH



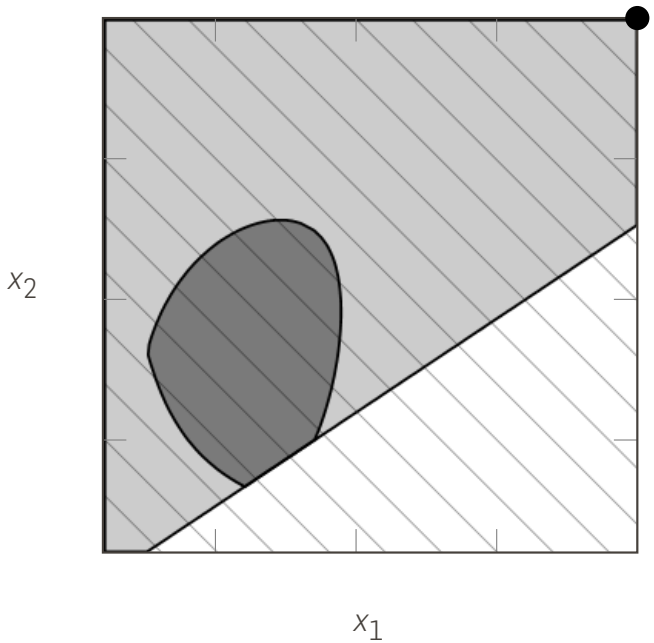
If an interior point is not given, obtain a feasible, relaxed interior point (satisfying all the nonlinear constraints in C) by solving a NLP problem.

LP STEP (OPTIONAL)



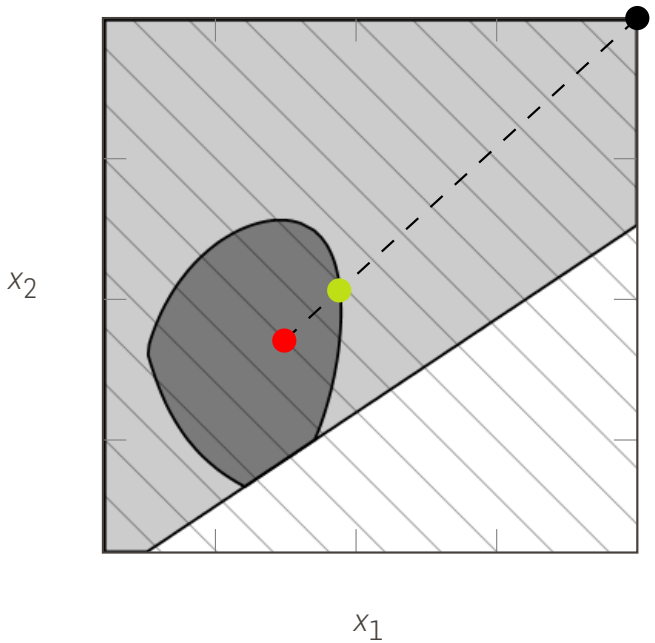
Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)



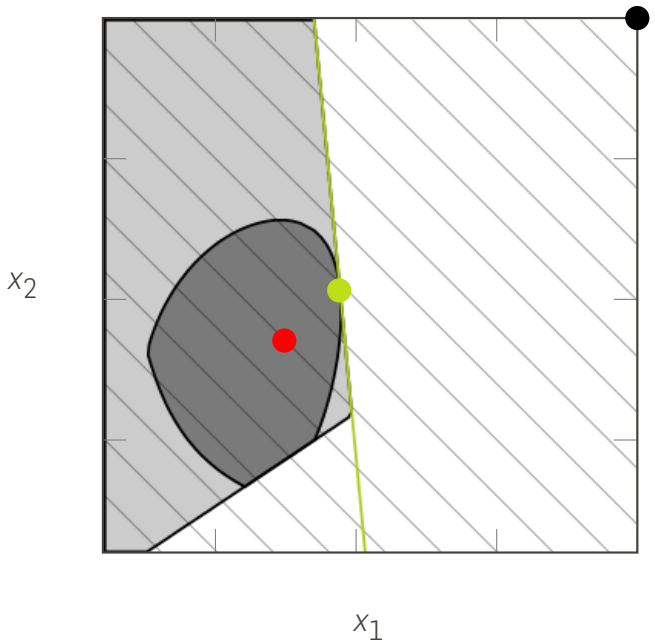
Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)



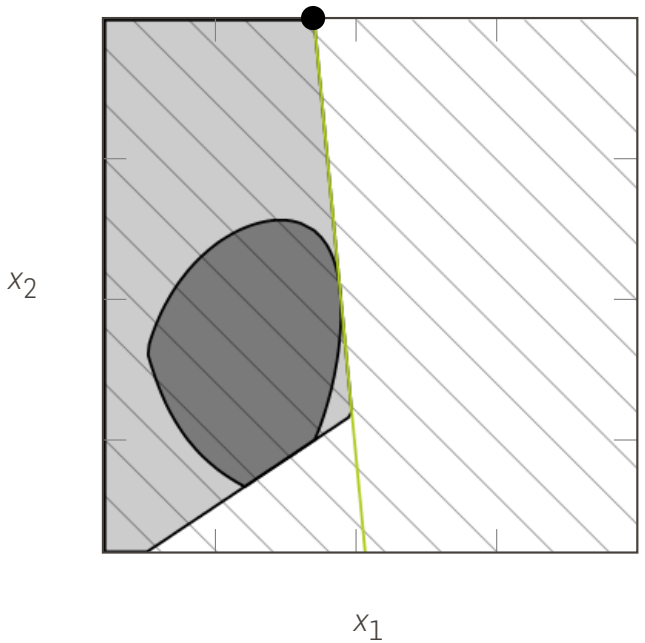
Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)



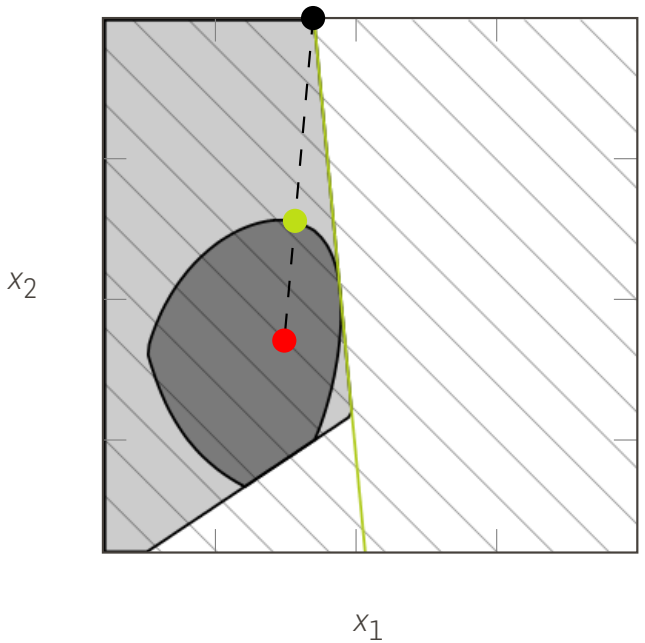
Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)



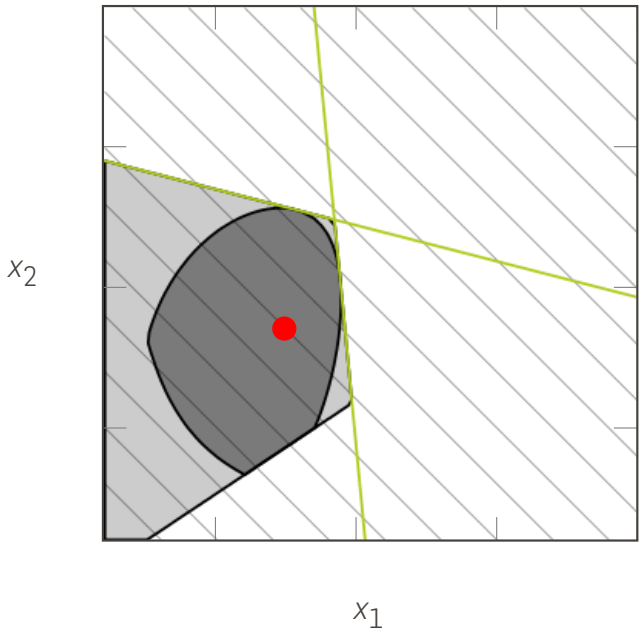
Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)



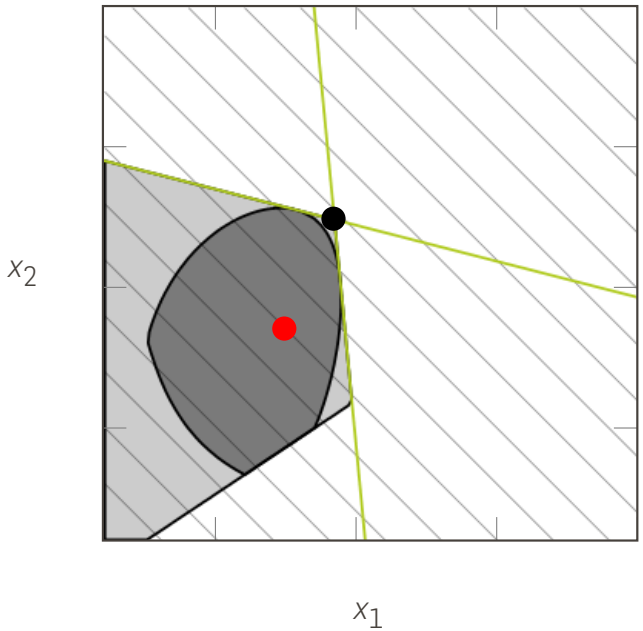
Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)



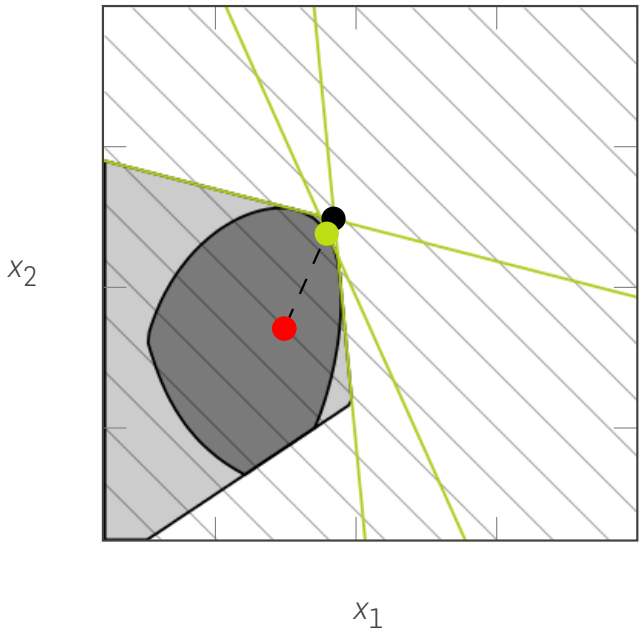
Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)



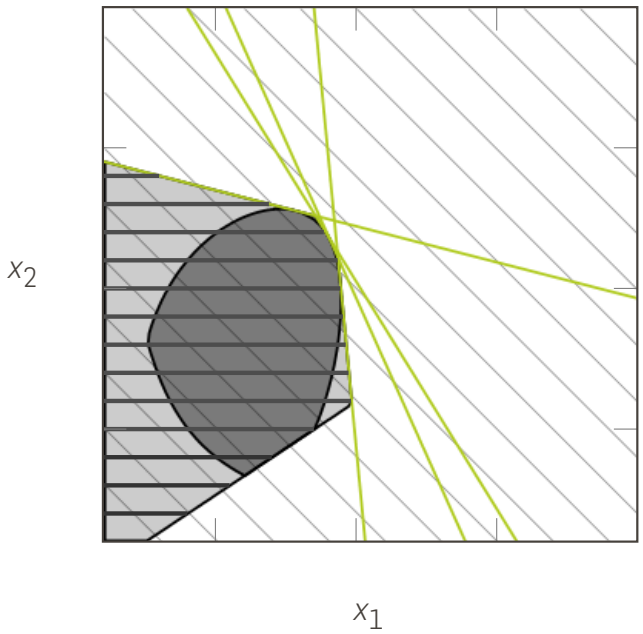
Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)



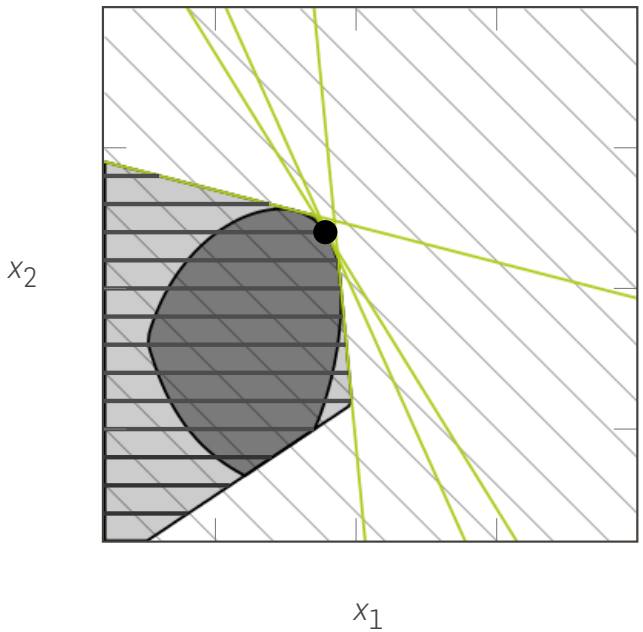
Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

MILP STEP



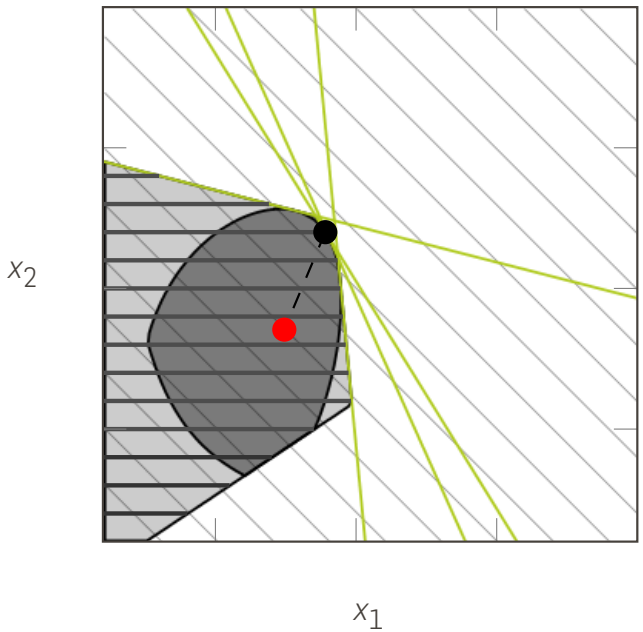
Finally include the integer requirements and solve MILP problems using a corresponding procedure to find the optimal solution to (P).

MILP STEP



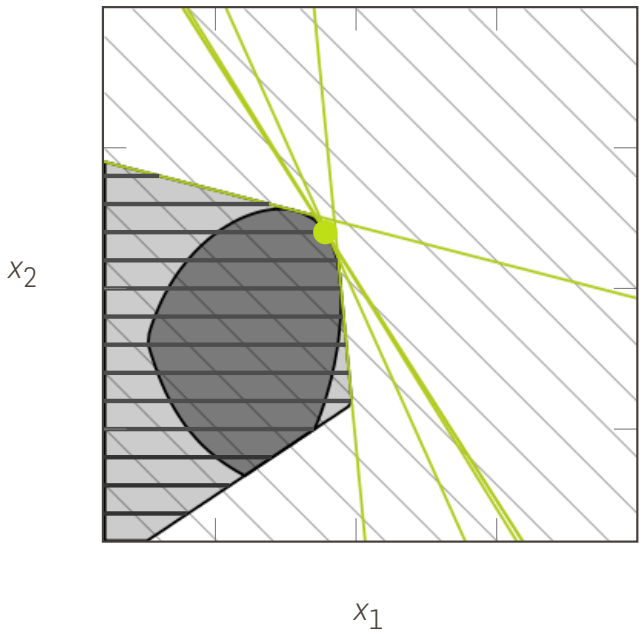
Finally include the integer requirements and solve MILP problems using a corresponding procedure to find the optimal solution to (P).

MILP STEP



Finally include the integer requirements and solve MILP problems using a corresponding procedure to find the optimal solution to (P).

MILP STEP



Finally include the integer requirements and solve MILP problems using a corresponding procedure to find the optimal solution to (P).

THE SUPPORTING HYPERPLANE OPTIMIZATION TOOLKIT SOLVER

THE SUPPORTING HYPERPLANE OPTIMIZATION TOOLKIT SOLVER

SHOT is an implementation of the ESH algorithm in C++ together with primal heuristics

THE SUPPORTING HYPERPLANE OPTIMIZATION TOOLKIT SOLVER

SHOT is an implementation of the ESH algorithm in C++ together with primal heuristics

- utilizes several COIN-OR subprojects: Optimization Services (OS), Ipopt, CBC

THE SUPPORTING HYPERPLANE OPTIMIZATION TOOLKIT SOLVER

SHOT is an implementation of the ESH algorithm in C++ together with primal heuristics

- utilizes several COIN-OR subprojects:
  Optimization Services (OS), Ipopt, CBC

- uses CPLEX, Gurobi, CBC for solving subproblems

THE SUPPORTING HYPERPLANE OPTIMIZATION TOOLKIT SOLVER

SHOT is an implementation of the ESH algorithm in C++ together with primal heuristics

- utilizes several COIN-OR subprojects: Optimization Services (OS), Ipopt, CBC
- uses CPLEX, Gurobi, CBC for solving subproblems
- uses Ipopt for finding the interior point (will also include the modified Kelley's method)

THE SUPPORTING HYPERPLANE OPTIMIZATION TOOLKIT SOLVER

SHOT is an implementation of the ESH algorithm in C++ together with primal heuristics

- utilizes several COIN-OR subprojects: Optimization Services (OS), Ipopt, CBC
- uses CPLEX, Gurobi, CBC for solving subproblems
- uses Ipopt for finding the interior point (will also include the modified Kelley's method)

SHOT will be released as an open source solver in COIN-OR.

FILE FORMATS SUPPORTED

SHOT reads problems in OSiL (Optimization Services instance Language) format and NL format.

```
<instanceData>
  <variables numberOfVariables="8">
    <var name="x1" />
    <var name="x2" />
    <var name="x3" />
    <var name="x4" />
    <var name="b6" type="B" ub="1" />
    <var name="b7" type="B" ub="1" />
    <var name="b8" type="B" ub="1" />
    <var name="b9" type="B" ub="1" />
  </variables>
  <objectives numberOfObjectives="1">
    <obj maxOrMin="min" name="defObj_objvar" numberOfObjCoef="0">
    </obj>
  </objectives>
  <constraints numberOfConstraints="7">
    <con name="e1" lb="1" ub="1" />
    <con name="e2" lb="10" ub="10" />
    <con name="e4" ub="0" />
    <con name="e5" ub="0" />
    <con name="e6" ub="0" />
    <con name="e7" ub="0" />
    <con name="e8" ub="3" />
  </constraints>
  <linearConstraintCoefficients numberOfValues="20">
    <start>
      <el mult="3" incr="4">0</el>
      <el mult="4" incr="2">10</el>
      <el>20</el>
    </start>
    <colIdx>
      <el mult="4" incr="1">0</el>
      <el mult="4" incr="1">0</el>
      <el>0</el>
```

FILE FORMATS SUPPORTED

SHOT reads problems in OSiL (Optimization Services instance Language) format and NL format.

Options are specified in OSoL (Optimization Services options Language) format.

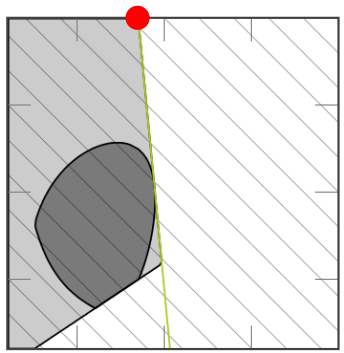
Results are provided in OSrL (Optimization Services results Language) format.

OSiL, OSoL and OSrL are XML-based and part of the Optimization Services project.

DUAL AND PRIMAL SOLUTIONS

A dual solution provides the best known lower bound on the objective value:

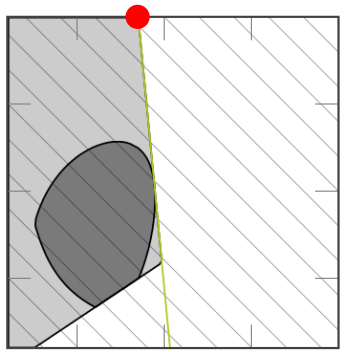
- Belongs to the relaxed set Ω_k.
- Provided by optimal solutions to the LP/MILP subproblems.



DUAL AND PRIMAL SOLUTIONS

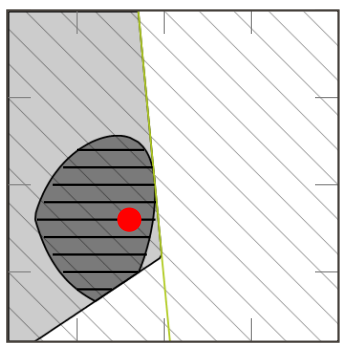
A dual solution provides the best known lower bound on the objective value:

- Belongs to the relaxed set Ω_k.
- Provided by optimal solutions to the LP/MILP subproblems.



A primal solution provides the best known integer-feasible solution to the MINLP problem:

- Belongs to the feasible set $C \cap L \cap Y$.
- Provided by primal heuristics.



DUAL SOLUTIONS (LOWER BOUND)

The dual solutions to the MINLP problem is given as the solutions to the LP/MILP subproblems.

- Only valid if the problems are solved to optimality
  - LP solutions valid if found
  - MILP solutions valid only if flagged optimal by the subsolver

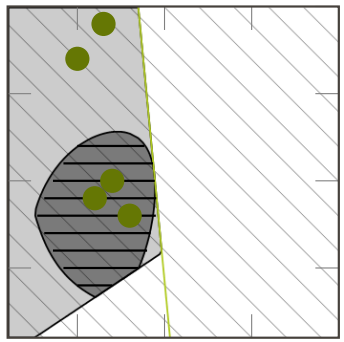
DUAL SOLUTIONS (LOWER BOUND)

The dual solutions to the MINLP problem is given as the solutions to the LP/MILP subproblems.

- Only valid if the problems are solved to optimality
  - LP solutions valid if found
  - MILP solutions valid only if flagged optimal by the subsolver

Many MILP solvers support a solution pool, *i.e.*, can return several feasible solutions

- these can be used to create more hyperplanes in each iteration



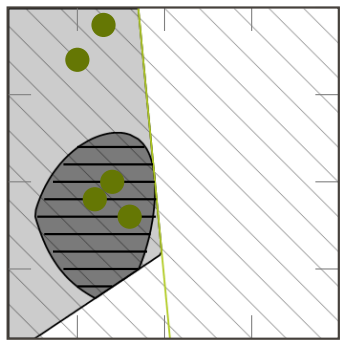
DUAL SOLUTIONS (LOWER BOUND)

The dual solutions to the MINLP problem is given as the solutions to the LP/MILP subproblems.

- Only valid if the problems are solved to optimality
  - LP solutions valid if found
  - MILP solutions valid only if flagged optimal by the subsolver

Many MILP solvers support a solution pool, *i.e.*, can return several feasible solutions

- these can be used to create more hyperplanes in each iteration



BUT… Too many hyperplanes may make each subsequent iteration computationally more expensive.

PRIMAL SOLUTIONS (UPPER BOUND)

Several techniques are used to find primal solutions:

- If points in the MILP solution pool are also in C these are primal solutions.
- Fix the integer-variables to the integer-valid solution and solve an NLP problem.

PRIMAL SOLUTIONS (UPPER BOUND)

Several techniques are used to find primal solutions:

- If points in the MILP solution pool are also in C these are primal solutions.
- Fix the integer-variables to the integer-valid solution and solve an NLP problem.

The primal solutions can be used for warm starts in the MILP solver as starting points or cut off values on the objective.

PRIMAL SOLUTIONS (UPPER BOUND)

Several techniques are used to find primal solutions:

- If points in the MILP solution pool are also in C these are primal solutions.
- Fix the integer-variables to the integer-valid solution and solve an NLP problem.

The primal solutions can be used for warm starts in the MILP solver as starting points or cut off values on the objective.

Future work

- Include more primal heuristics in SHOT, *e.g.*, based on line searches.

PRIMAL SOLUTIONS FROM THE MILP SOLUTION POOL

Several parameters available in the MILP solver that determine the returned solutions in the solution pool

`MIPEmphasis`, `Probe`, `SolnPoolGap`, `SolnPoolIntensity`…

PRIMAL SOLUTIONS FROM THE MILP SOLUTION POOL

Several parameters available in the MILP solver that determine the returned solutions in the solution pool

`MIPEmphasis`, `Probe`, `SolnPoolGap`, `SolnPoolIntensity`…

It is also possible to use the populate functionality of the MILP solvers to find additional solutions.

» Makes the solution process nondeterministic in many cases if time limit is used in subsolver.

PRIMAL SOLUTIONS FROM THE MILP SOLUTION POOL

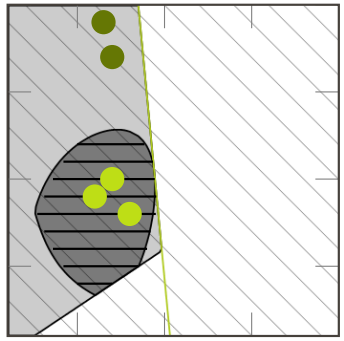
Several parameters available in the MILP solver that determine the returned solutions in the solution pool

`MIPEmphasis`, `Probe`, `SolnPoolGap`, `SolnPoolIntensity`…

It is also possible to use the populate functionality of the MILP solvers to find additional solutions.

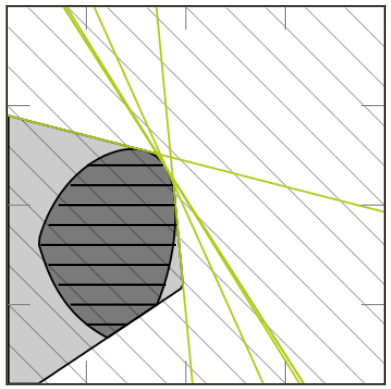
» Makes the solution process nondeterministic in many cases if time limit is used in subsolver.

Often it is better to obtain points with ‘bad’ objective values in the linear relaxation for the primal solutions.



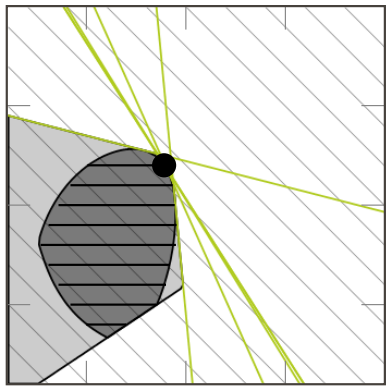
PRIMAL SOLUTIONS FROM SOLVING NLP PROBLEMS

To obtain a primal solution, it is possible to fix the integer-values of an integer-feasible solution (not necessarily in $C \cap L$) and solve an NLP problem.



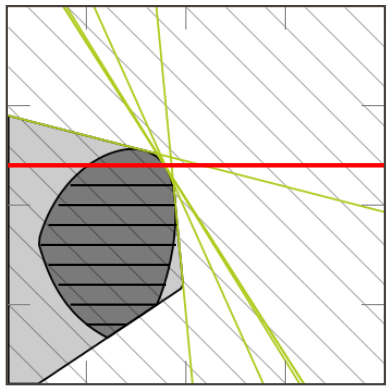
PRIMAL SOLUTIONS FROM SOLVING NLP PROBLEMS

To obtain a primal solution, it is possible to fix the integer-values of an integer-feasible solution (not necessarily in $C \cap L$) and solve an NLP problem.



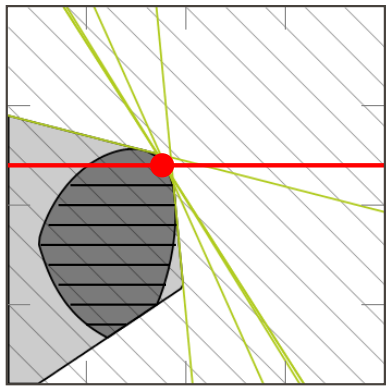
PRIMAL SOLUTIONS FROM SOLVING NLP PROBLEMS

To obtain a primal solution, it is possible to fix the integer-values of an integer-feasible solution (not necessarily in $C \cap L$) and solve an NLP problem.



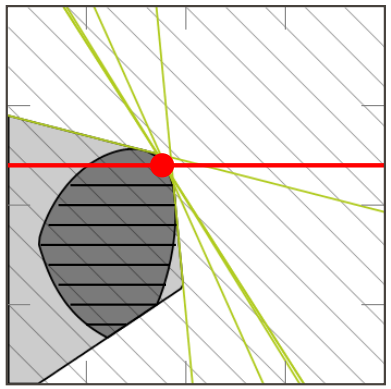
PRIMAL SOLUTIONS FROM SOLVING NLP PROBLEMS

To obtain a primal solution, it is possible to fix the integer-values of an integer-feasible solution (not necessarily in $C \cap L$) and solve an NLP problem.



PRIMAL SOLUTIONS FROM SOLVING NLP PROBLEMS

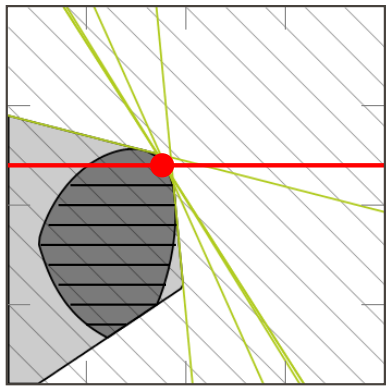
To obtain a primal solution, it is possible to fix the integer-values of an integer-feasible solution (not necessarily in $C \cap L$) and solve an NLP problem.



The NLP problem is not always feasible!

PRIMAL SOLUTIONS FROM SOLVING NLP PROBLEMS

To obtain a primal solution, it is possible to fix the integer-values of an integer-feasible solution (not necessarily in $C \cap L$) and solve an NLP problem.



The NLP problem is not always feasible!

In SHOT, Ipopt is used to solve the NLP problem:

- Strategy executed at a specific iteration or time interval
- If the solution is on the boundary of the feasible region or outside (to an ϵ tolerance), a supporting hyperplane is generated.

TERMINATION CRITERIA IN SHOT

Absolute or relative objective duality gap

$$|\mathrm{DB} - \mathrm{PB}| \leq \epsilon_{\mathrm{abs}} \qquad \frac{|\mathrm{DB} - \mathrm{PB}|}{10^{-10} + |\mathrm{PB}|} \leq \epsilon_{\mathrm{rel}}$$

where DB and PB are the dual and primal objective values

TERMINATION CRITERIA IN SHOT

Absolute or relative objective duality gap

$$|\mathrm{DB} - \mathrm{PB}| \leq \epsilon_{\mathrm{abs}} \qquad \frac{|\mathrm{DB} - \mathrm{PB}|}{10^{-10} + |\mathrm{PB}|} \leq \epsilon_{\mathrm{rel}}$$

where DB and PB are the dual and primal objective values

Constraint feasibility tolerance

$$\max_m g_m(x^k_{\mathrm{MILP}}) \leq \epsilon_{\mathrm{MILP}}$$

TERMINATION CRITERIA IN SHOT

Absolute or relative objective duality gap

$$|\mathrm{DB} - \mathrm{PB}| \leq \epsilon_{\mathrm{abs}} \qquad \frac{|\mathrm{DB} - \mathrm{PB}|}{10^{-10} + |\mathrm{PB}|} \leq \epsilon_{\mathrm{rel}}$$

where DB and PB are the dual and primal objective values

Constraint feasibility tolerance

$$\max_m g_m(x^k_{\mathrm{MILP}}) \leq \epsilon_{\mathrm{MILP}}$$

Iteration or time limit reached

QUADRATIC AND NONLINEAR OBJECTIVE FUNCTIONS

Normally, a nonlinear objective function $f(x)$ is rewritten as a constraint

$$f(x) - \mu \leq 0$$

and the new objective is to minimize the auxiliary variable μ.

QUADRATIC AND NONLINEAR OBJECTIVE FUNCTIONS

Normally, a nonlinear objective function $f(x)$ is rewritten as a constraint

$$f(x) - \mu \leq 0$$

and the new objective is to minimize the auxiliary variable μ.

Many MILP solvers can directly solve MIQP problems:

- Then $f(x) = x^T Q x + c^T x$, where Q positive semidefinite.
- Gives the exact objective instead of a linearization.

QUADRATIC AND NONLINEAR OBJECTIVE FUNCTIONS

Normally, a nonlinear objective function $f(x)$ is rewritten as a constraint

$$f(x) - \mu \leq 0$$

and the new objective is to minimize the auxiliary variable μ.

Many MILP solvers can directly solve MIQP problems:

- Then $f(x) = x^T Q x + c^T x$, where Q positive semidefinite.
- Gives the exact objective instead of a linearization.

Quadratic constraints can also be handled by some solvers.

- Numerical issues may lead to trouble with proving semidefiniteness.
- Not always more effective than using the ESH algorithm for these constraints.

PERFORMANCE OF SHOT

SHOT was tested on all 333 MINLP instances classified as convex in the MINLP Library 2:

- Number of variables in the problems 3 – 107 223 (mean 999).
- Largest number of discrete variables in a problem is 1500.
- All benchmarks performed on Linux-based 64 bit computer (Intel Xeon 3.6 GHz, four physical and eight logical cores) with 32 GB RAM. Subsolvers used were CPLEX 12.6.1 and IPOPT 3.11.7.

PERFORMANCE OF SHOT

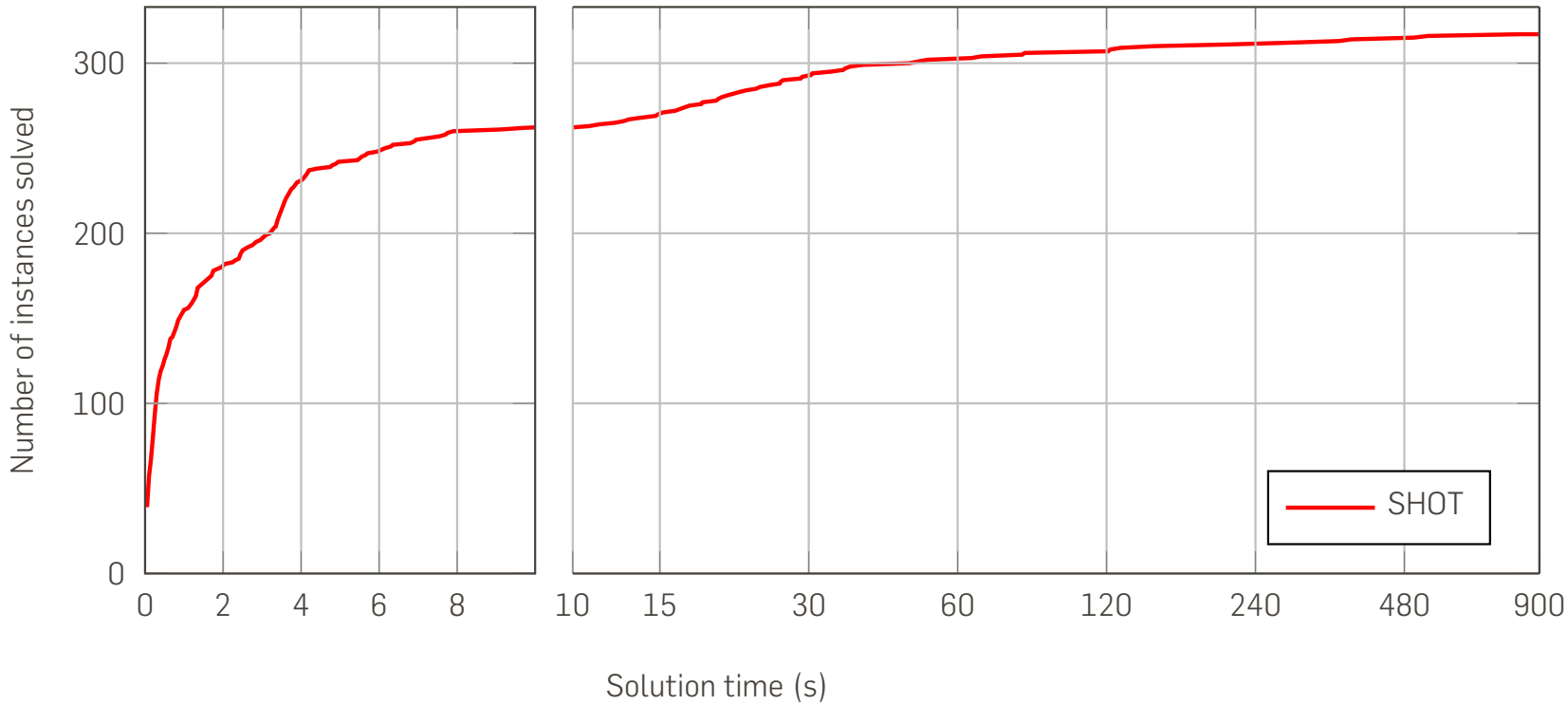
SHOT was tested on all 333 MINLP instances classified as convex in the MINLP Library 2:

- Number of variables in the problems 3 – 107 223 (mean 999).
- Largest number of discrete variables in a problem is 1500.
- All benchmarks performed on Linux-based 64 bit computer (Intel Xeon 3.6 GHz, four physical and eight logical cores) with 32 GB RAM. Subsolvers used were CPLEX 12.6.1 and IPOPT 3.11.7.

Solution strategy

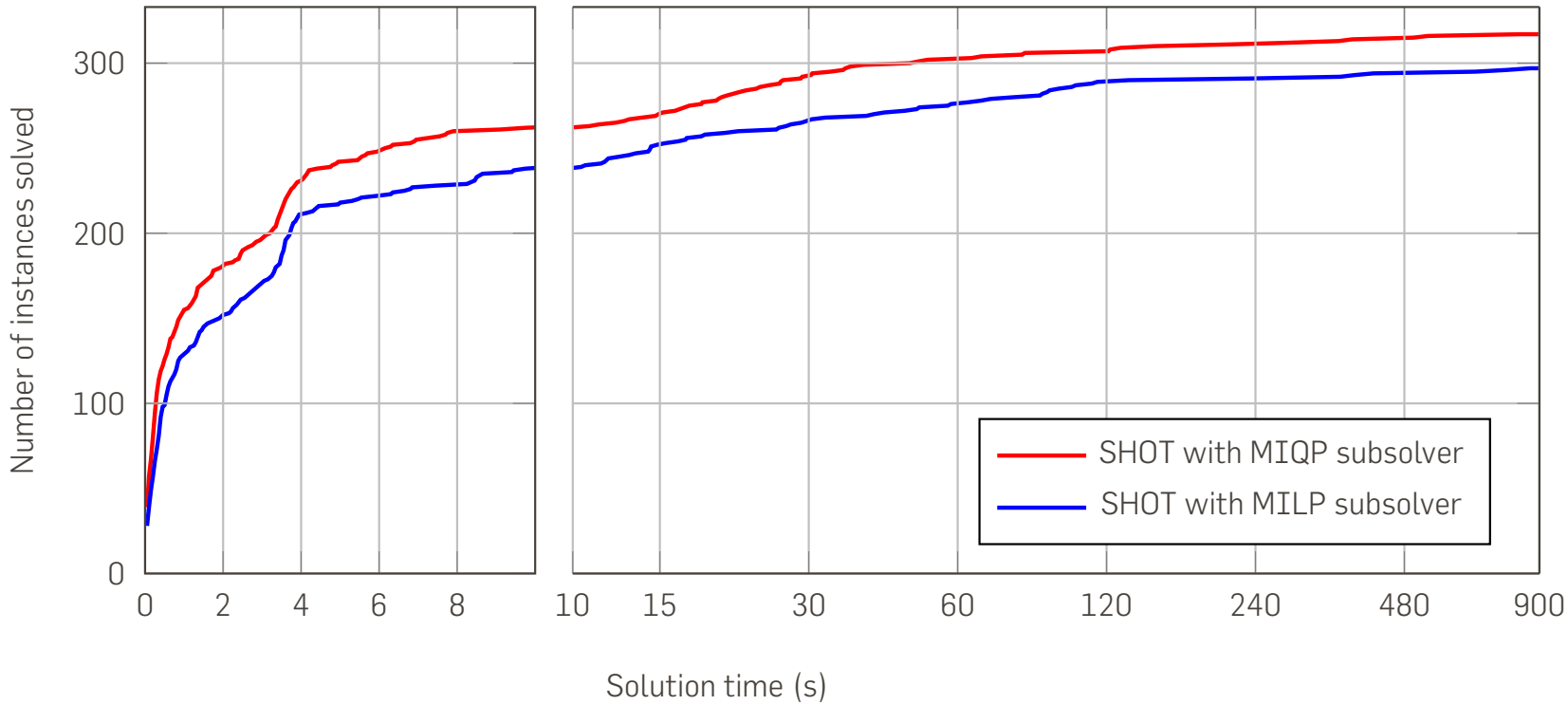
- $\epsilon_{\text{abs}} = \epsilon_{\text{rel}} = 0.001$, $\epsilon_{\text{MILP}} = 10^{-5}$, $\epsilon_{\text{LP}} = 0.001$, $K_{\text{LP}} = 300$
- maximal solution pool size: 10
- quadratic objective functions passed on to subsolver
- quadratic constraints regarded as general nonlinear

PERFORMANCE OF SHOT



A performance profile of the number of problem instances solved by SHOT to an objective duality gap $\leq 1\%$ as calculated by PAVER 2.

IMPACT OF USING THE QUADRATIC OBJECTIVE STRATEGY



A performance profile of the number of problem instances solved by SHOT to an objective duality gap $\leq 1\%$ with a MIQP or MILP subsolver.

SOLUTION LIMIT STRATEGY

It is possible to set the number of feasible solutions to find before terminating the MILP/MIQP subsolver.

SOLUTION LIMIT STRATEGY

It is possible to set the number of feasible solutions to find before terminating the MILP/MIQP subsolver.

Can be used to speed up the initial MILP iterations:

- Optimal solution of intermediate subproblems not required.
- Reduces solution time significantly in many cases.
- If no constraints are added to the subproblem, the solution limit can be increased without rebuilding the branching tree.

SOLUTION LIMIT STRATEGY

It is possible to set the number of feasible solutions to find before terminating the MILP/MIQP subsolver.

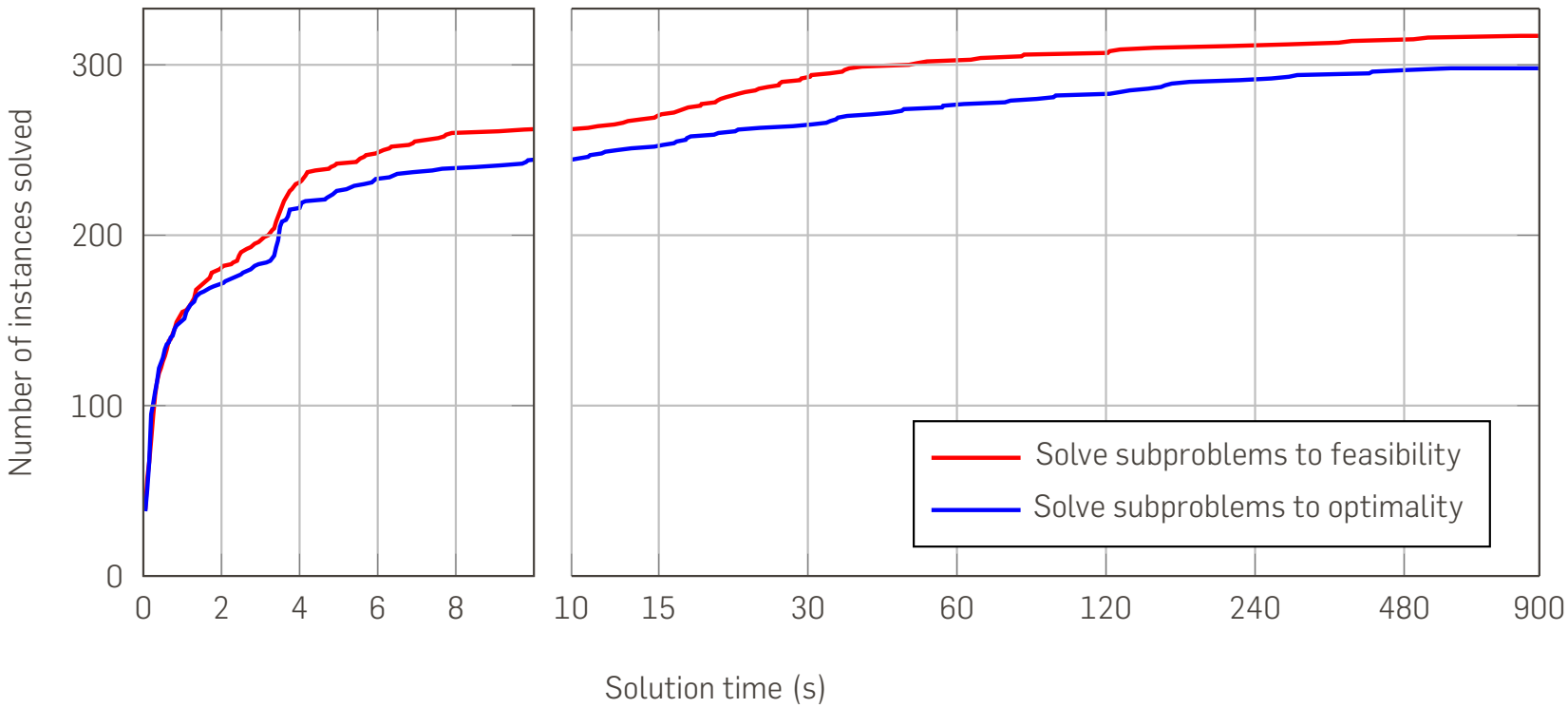
Can be used to speed up the initial MILP iterations:

- Optimal solution of intermediate subproblems not required.
- Reduces solution time significantly in many cases.
- If no constraints are added to the subproblem, the solution limit can be increased without rebuilding the branching tree.

Solution limit strategy

1. Initially set SOLLIM $= 1$.
2. Solve MILP/MIQP subproblem and obtain solution x.
3. Terminate if x is MILP optimal and ESH termination criterion fulfilled.
4. Increase SOLLIM and goto step 2 if x is MILP optimal and $\max_m g_m(x) \leq \epsilon_{\text{SL}}$.
5. Add supporting hyperplanes and goto step 2.

SOLUTION LIMIT STRATEGY IMPACT



Performance profiles of the number of problem instances solved by SHOT to an objective duality gap $\leq 1\%$ with or without an increasing solution limit strategy.

LP RELAXATION STRATEGY

Initially integer-relaxed MILP/MIQP problems, *i.e.*, LP/QP problems, can be solved:

+ Integer-relaxed problems are much faster to solve.
– For some problems the hyperplanes generated may provide a bad relaxation.
– Hyperplanes generated may reduce overall performance for large problems.

LP RELAXATION STRATEGY

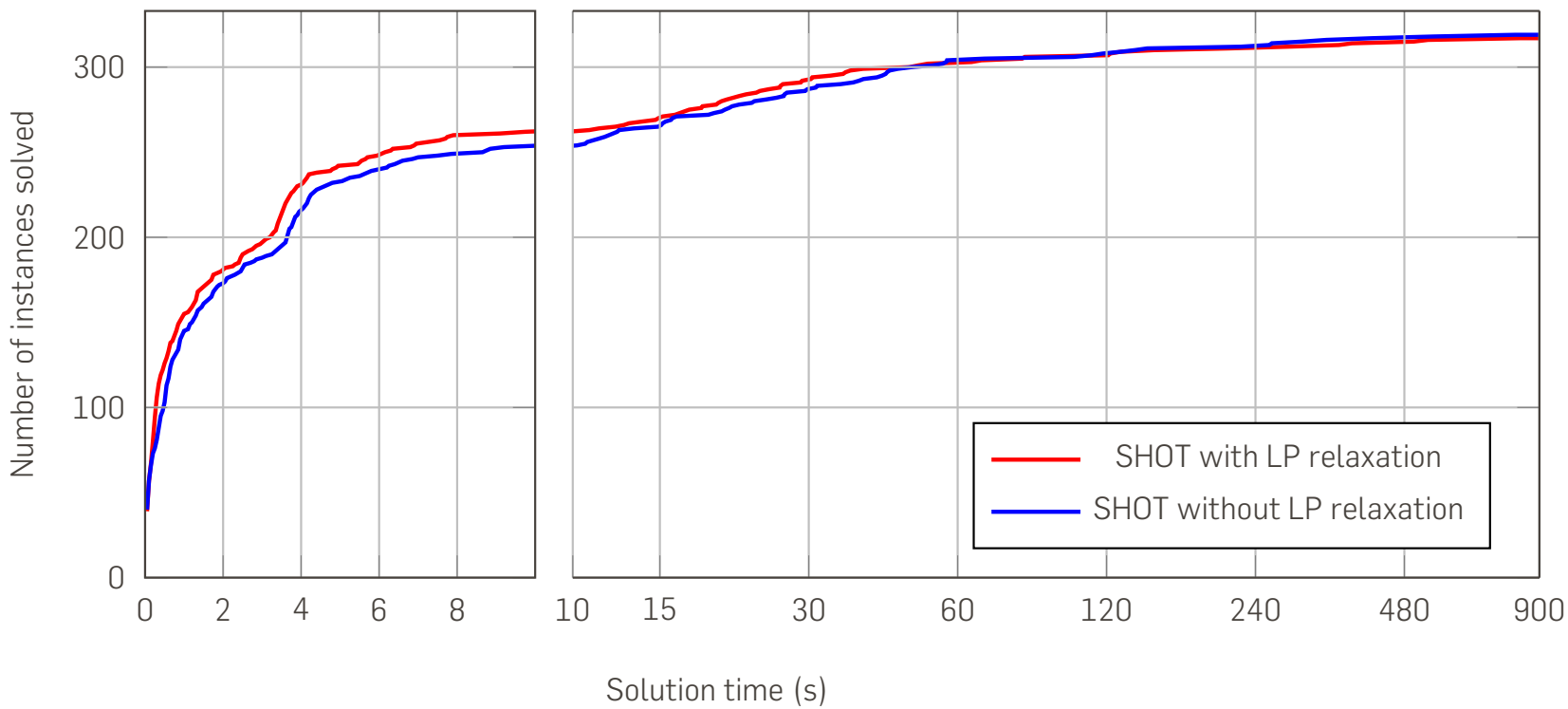
Initially integer-relaxed MILP/MIQP problems, *i.e.*, LP/QP problems, can be solved:

+ Integer-relaxed problems are much faster to solve.
– For some problems the hyperplanes generated may provide a bad relaxation.
– Hyperplanes generated may reduce overall performance for large problems.

Future work

» Investigate adding these supporting hyperplanes as lazy constraints.

LP RELAXATION STRATEGY IMPACT



Performance profiles of the number of problem instances solved by SHOT to an objective gap $\leq 1\%$ with or without a strategy for solving integer-relaxed problems.

BENCHMARKS AGAINST OTHER MINLP SOLVERS

BENCHMARKS AGAINST OTHER SOLVERS

The following MINLP solvers available in GAMS 24.4.1 were used:

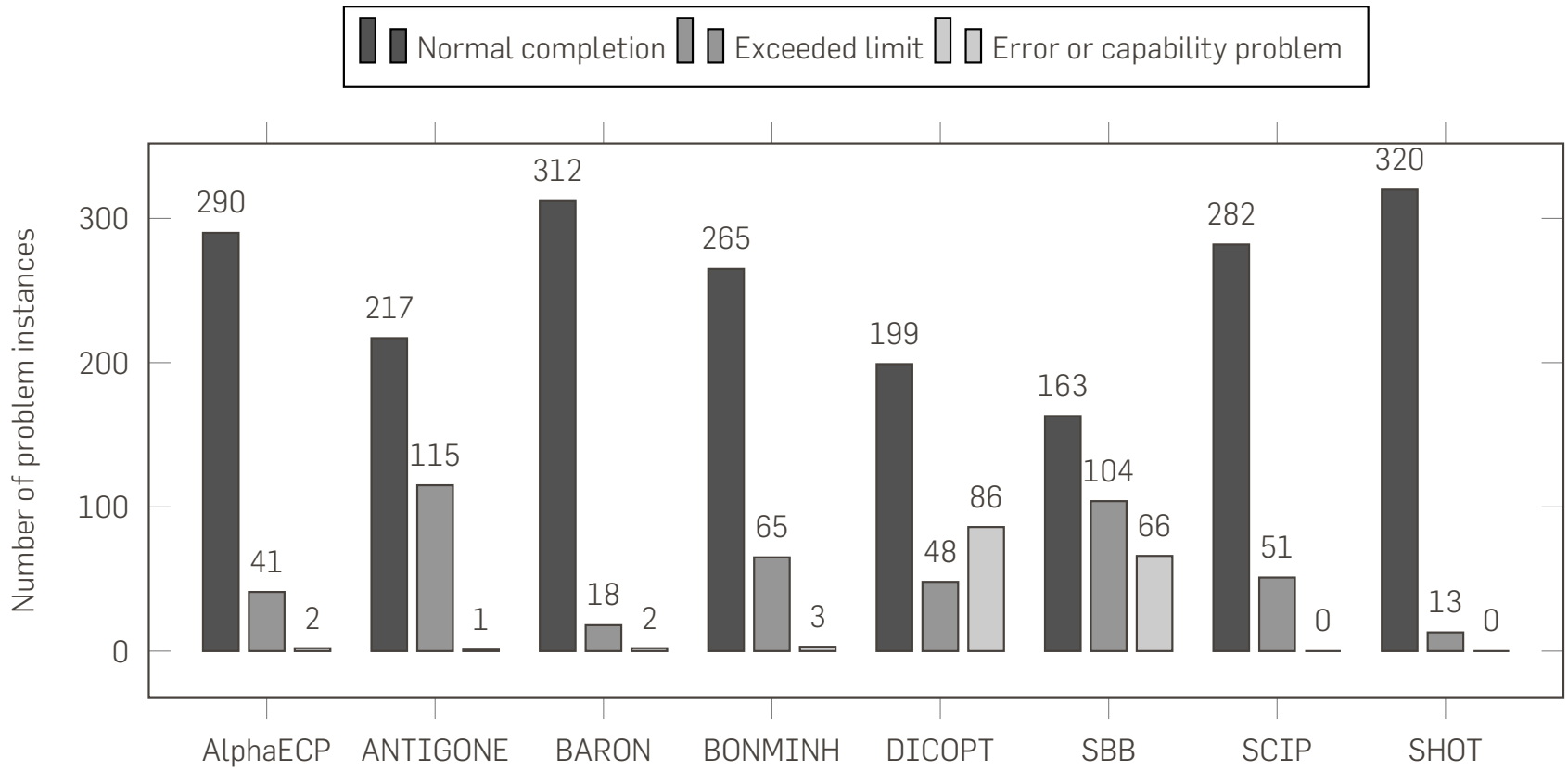
- AlphaECP (with convex strategy)
- ANTIGONE
- BARON
- BONMINH (with recommended convex strategy, B-Hyb)
- DICOPT
- SBB
- SCIP (with convex strategy)

CPLEX and CONOPT were used as subsolvers and the time limit per problem was 900 s.

The same problem set as before was used, *i.e.*, all 333 convex MINLP problems in MINLPLib 2.

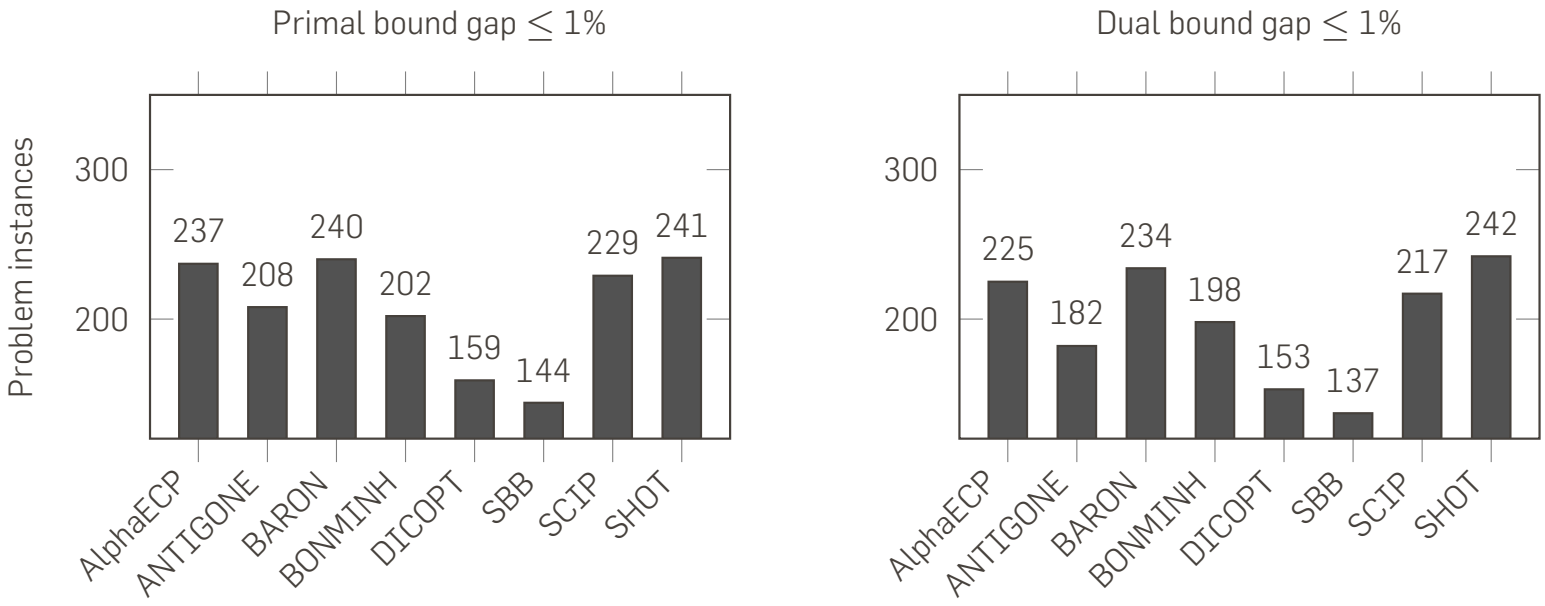
The total computational time for all solvers was about 150 h.

REASONS FOR TERMINATION



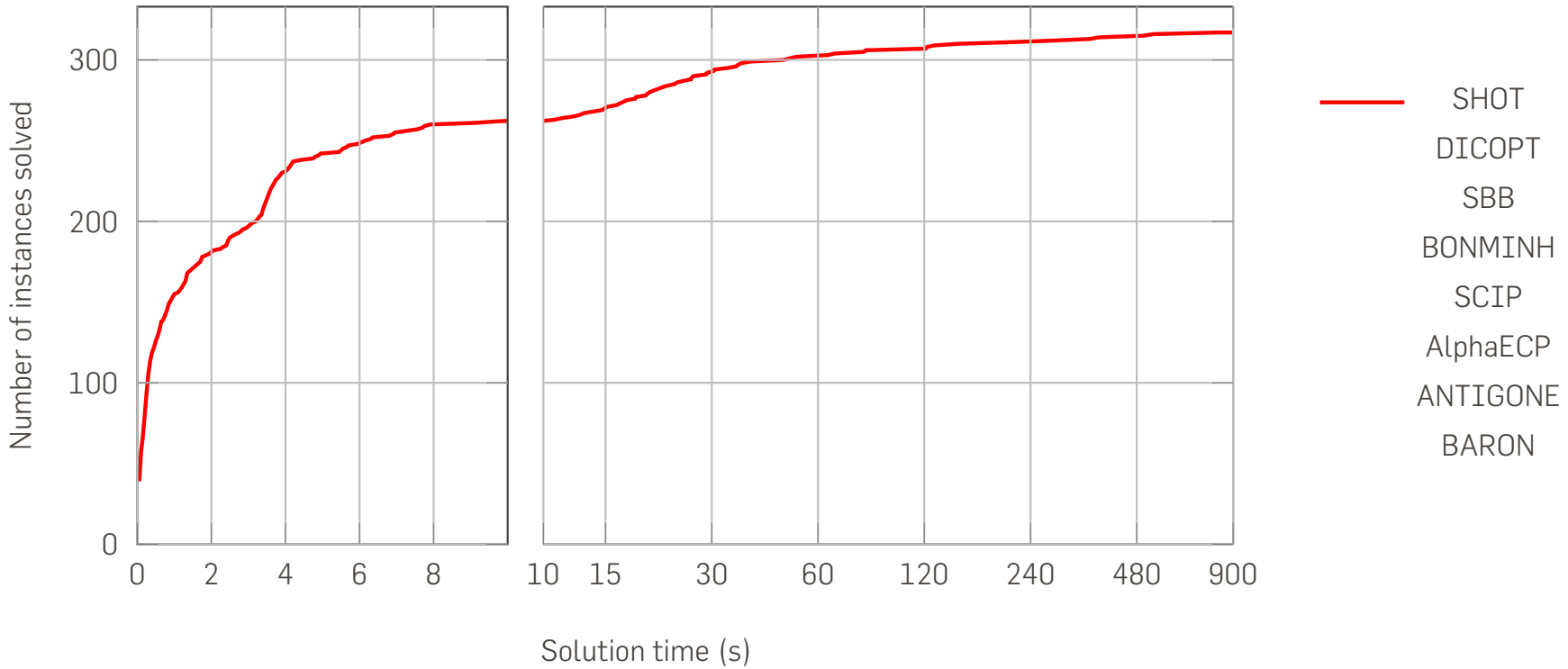
The termination statuses for the solvers as provided by PAVER 2. Note that a the solution can be optimal even though a limit (*e.g.*, time) has been reached, the solver has simply not proven optimality.

PRIMAL AND DUAL SOLUTION GAPS



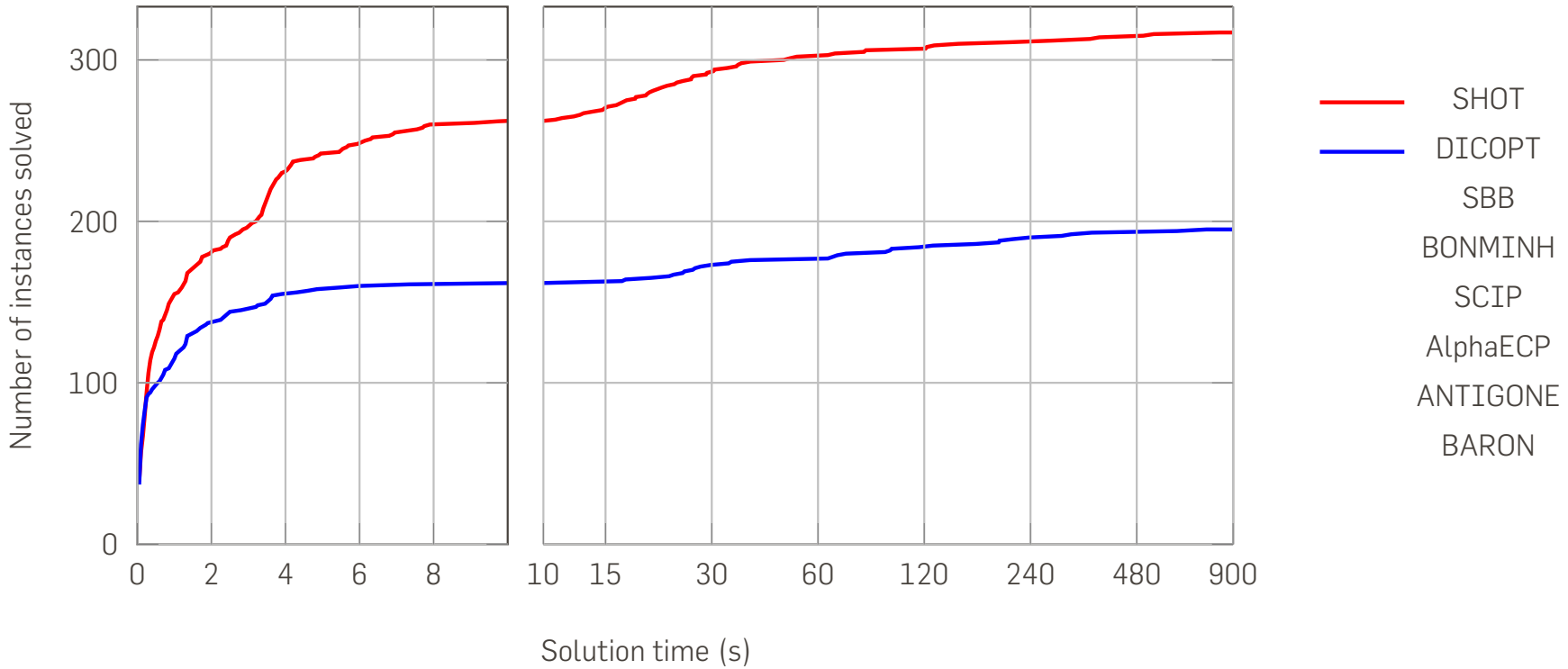
Number of problem instances solved with primal and dual solution bound gaps $\leq 1\%$ as calculated by PAVER 2 for the 242 problems with reported optimal solution available in MINLPLib 2.

PERFORMANCE PROFILE



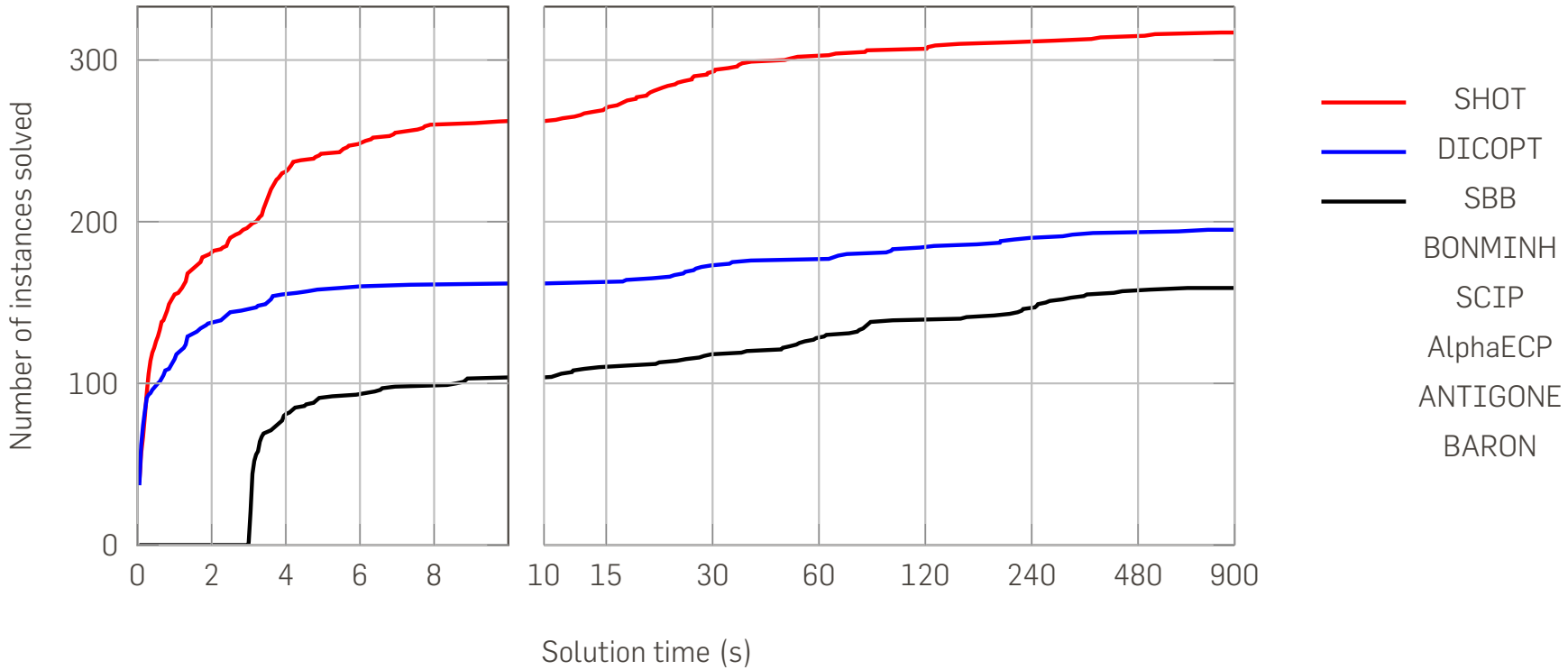
A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

PERFORMANCE PROFILE



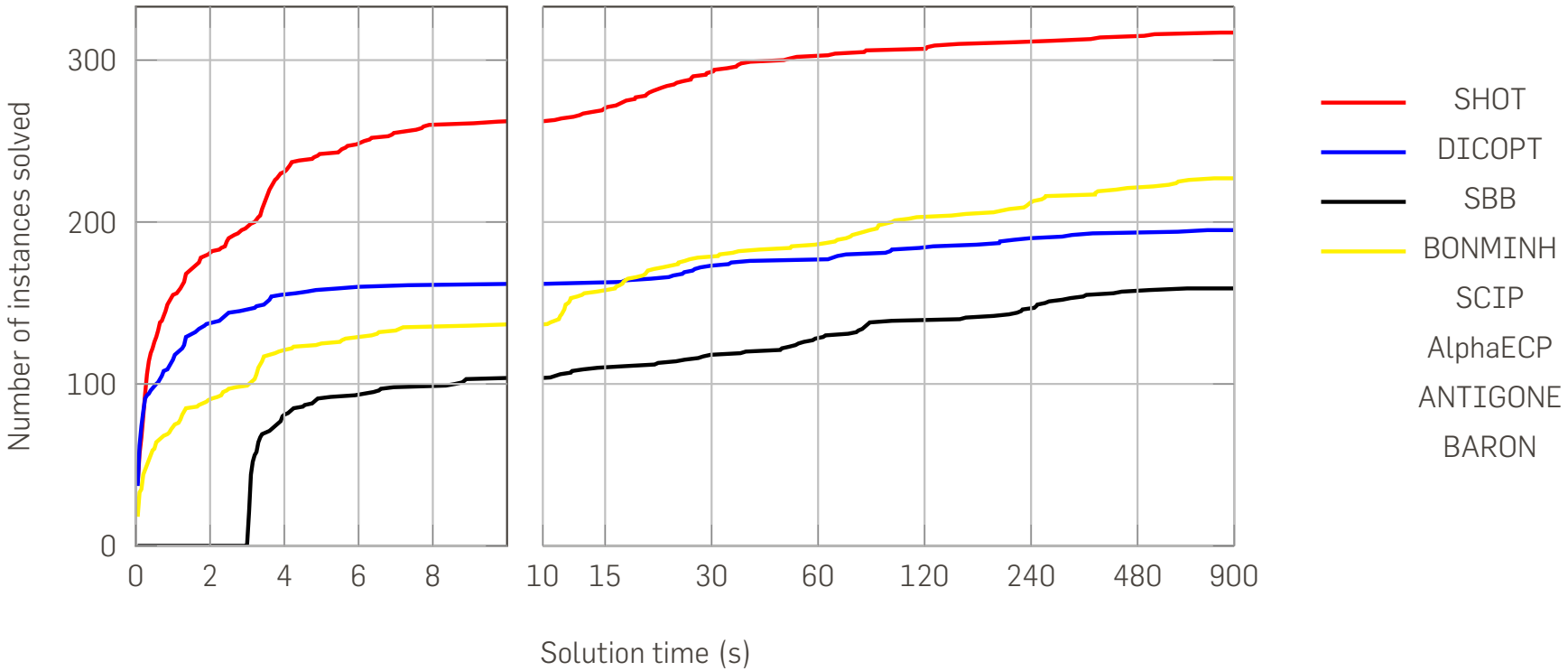
A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

PERFORMANCE PROFILE



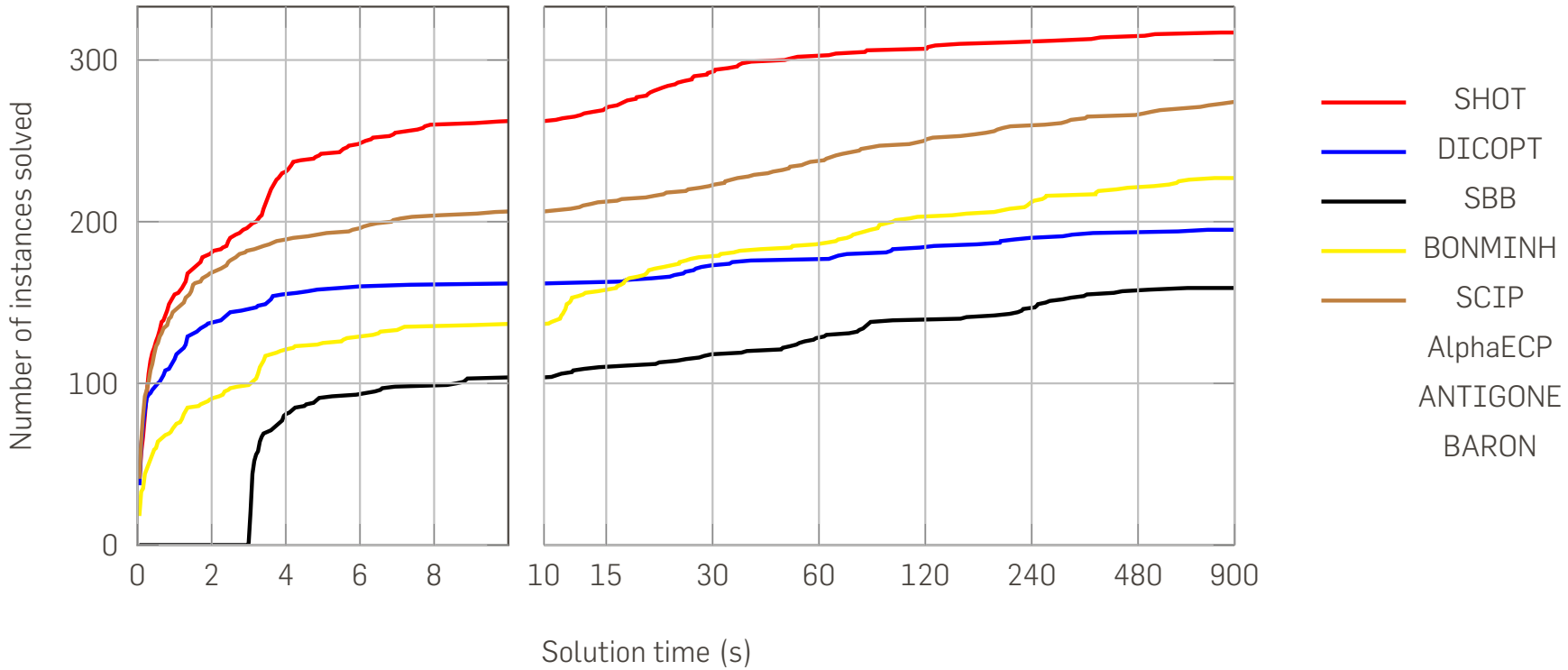
A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

PERFORMANCE PROFILE



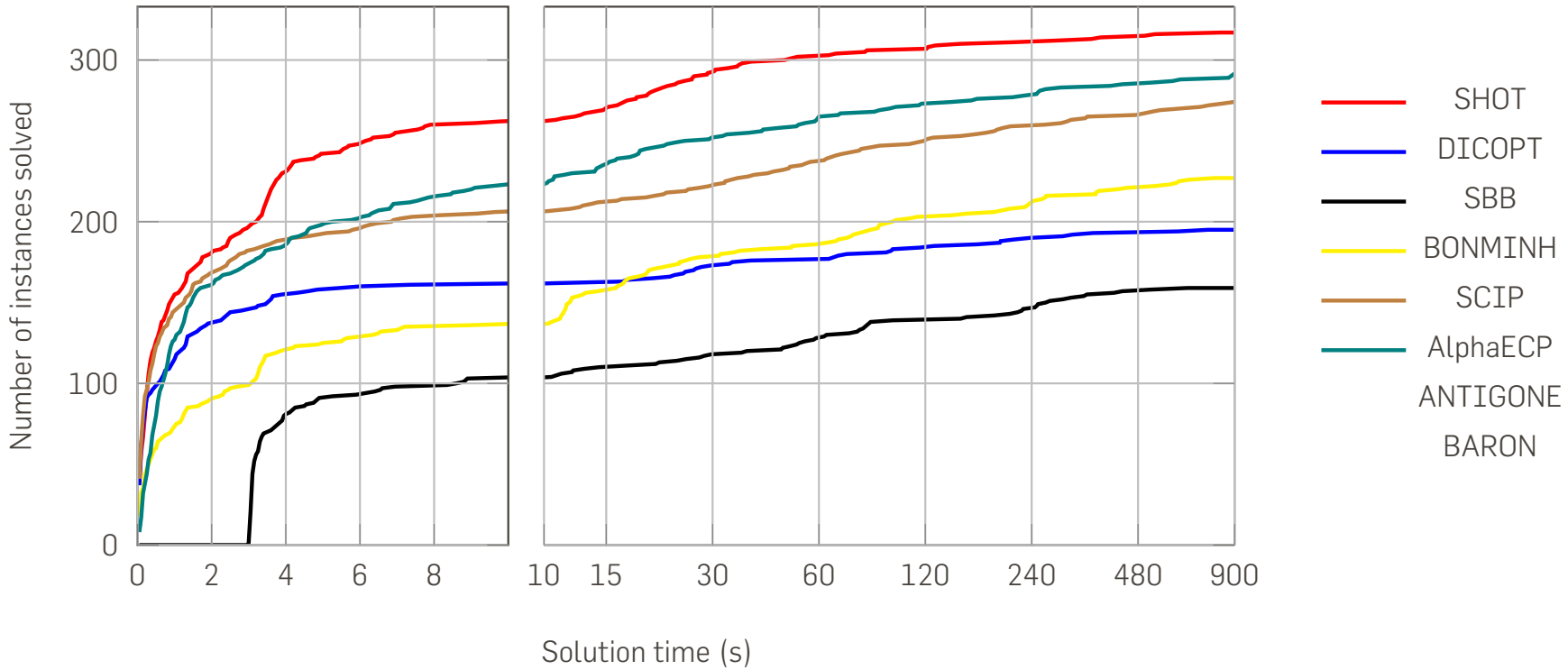
A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

PERFORMANCE PROFILE



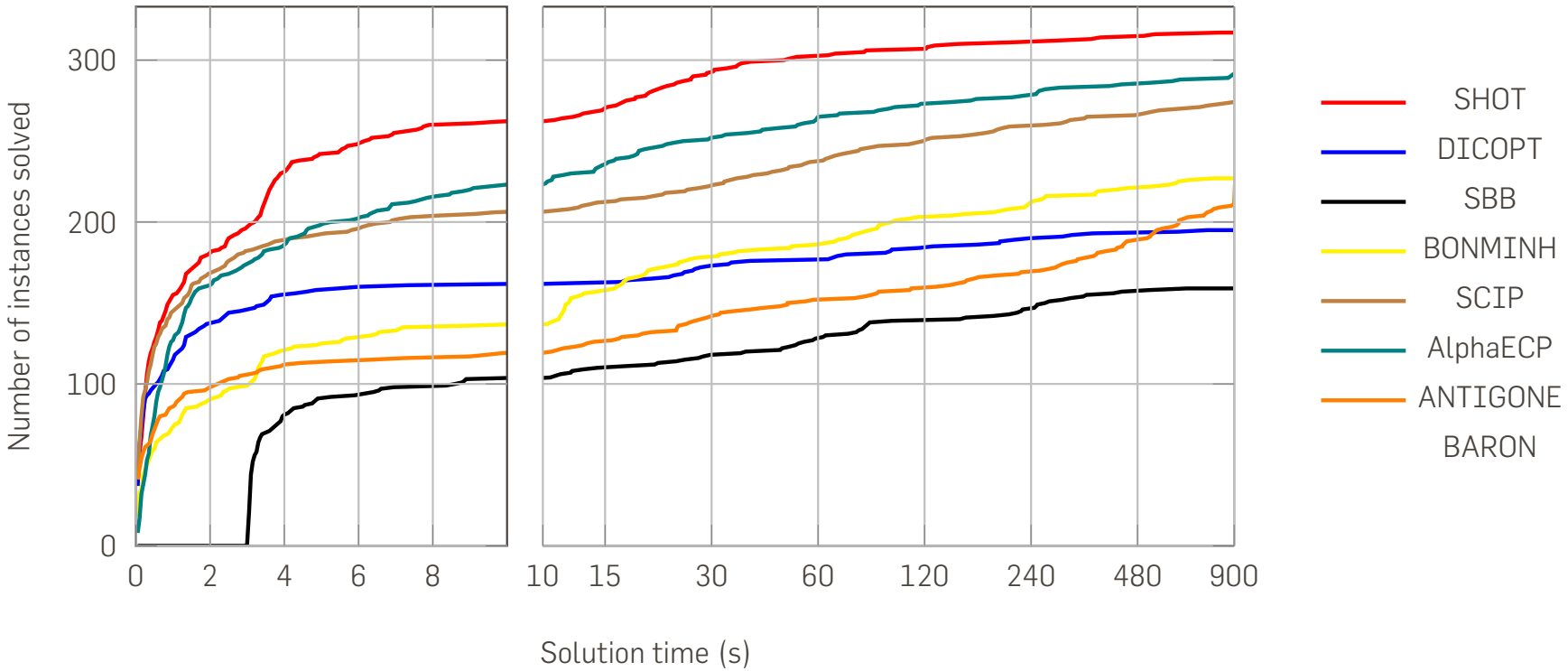
A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

PERFORMANCE PROFILE



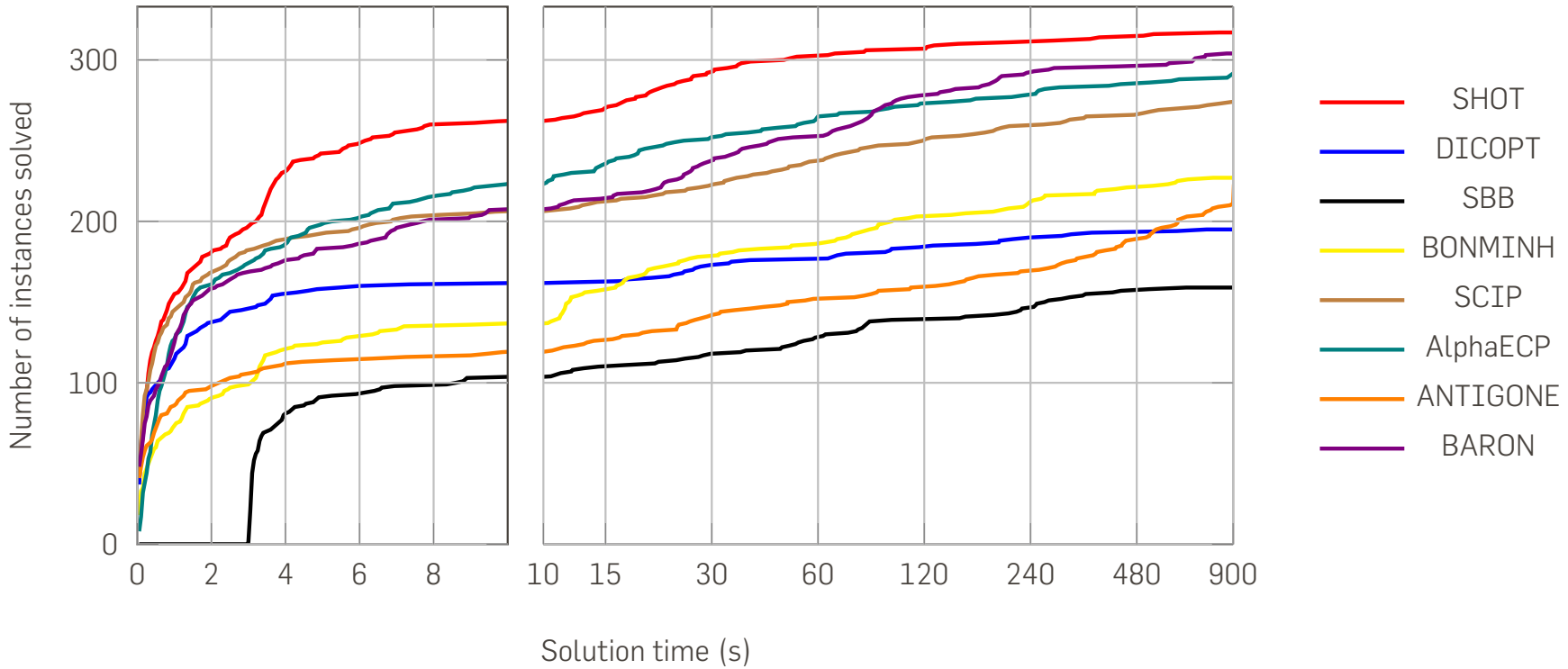
A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

PERFORMANCE PROFILE



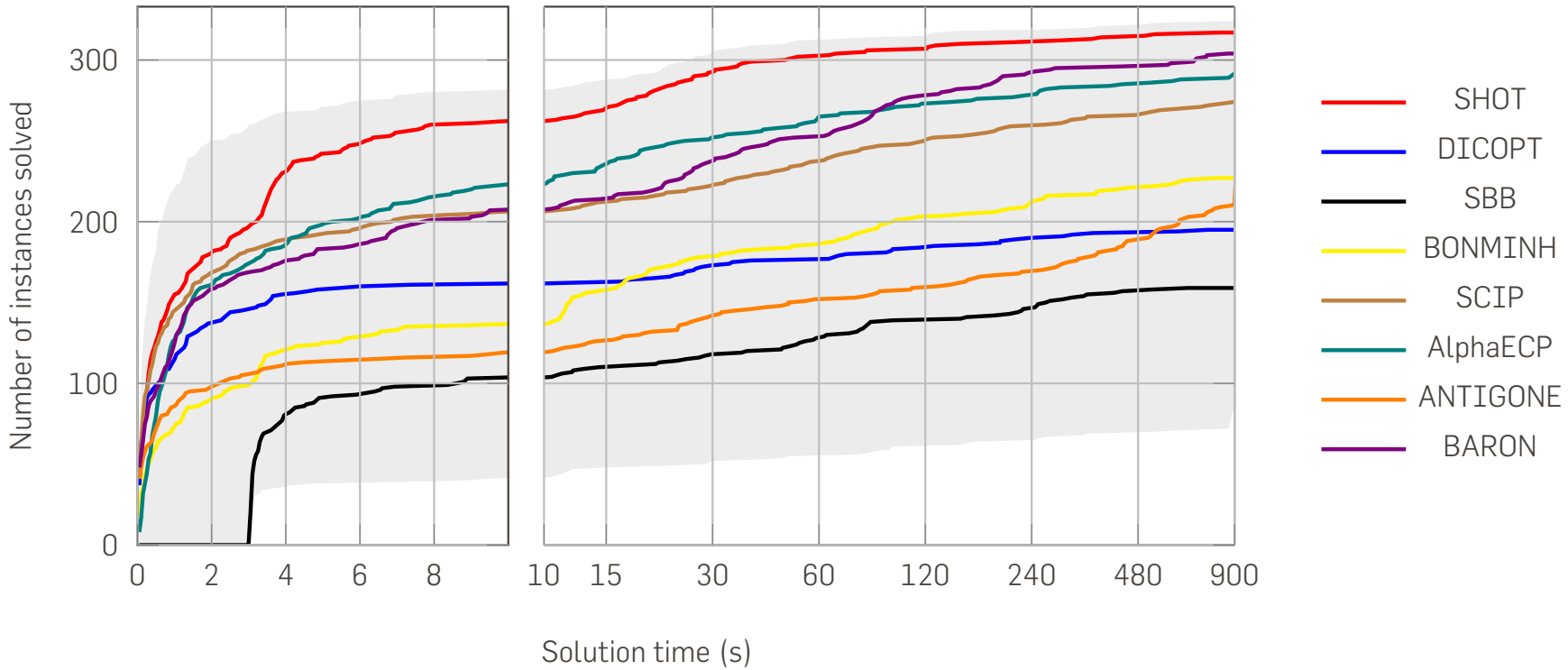
A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

PERFORMANCE PROFILE



A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

PERFORMANCE PROFILE



A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

CONCLUDING REMARKS

CONCLUDING REMARKS

The ESH algorithm is a new method for convex MINLP.

- A journal paper in Journal of Global Optimization will be available online *very* soon.

CONCLUDING REMARKS

The ESH algorithm is a new method for convex MINLP.

- A journal paper in Journal of Global Optimization will be available online *very* soon.

SHOT is an implementation of the ESH algorithm together with primal heuristics.

- SHOT will be released as an open source solver in COIN-OR (hopefully) during 2015.

CONCLUDING REMARKS

The ESH algorithm is a new method for convex MINLP.

- A journal paper in Journal of Global Optimization will be available online *very* soon.

SHOT is an implementation of the ESH algorithm together with primal heuristics.

- SHOT will be released as an open source solver in COIN-OR (hopefully) during 2015.

Future research and development

- Investigate how to best select the interior point
- Improve handling of nonlinear objective functions
- Include the αSGO framework for global optimization of nonconvex MINLP problems

Thank you for your attention!

Thank you for your attention!

Questions or comments?