IMPROVEMENTS TO THE SHOT SOLVER FOR CONVEX MINLP

XIII Global Optimization Workshop – GOW'16
4-8 September 2016, Braga, Portugal

Andreas Lundell, Jan Kronqvist and
Tapio Westerlund

Optimization and Systems Engineering
Åbo Akademi University, Finland

OPTIMIZATION AND SYSTEMS ENGINEERING

Åbo Akademi University

CONTENTS OF THE TALK

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

The **extended supporting hyperplane (ESH) algorithm** is an algorithm 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.

The **extended supporting hyperplane (ESH) algorithm** is an algorithm 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.

Enhancements increasing the computational efficiency of SHOT, *e.g.*, using **lazy constraints** are described.

CONTENTS OF THE TALK

The **extended supporting hyperplane (ESH) algorithm** is an algorithm 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.

Enhancements increasing the computational efficiency of SHOT, *e.g.,* using **lazy constraints** are described.

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, Vol. 64(2), pp. 249–272, 2016

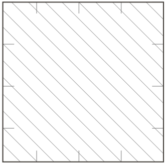
THE MINLP PROBLEM SCOPE

The ESH algorithm solves convex MINLP problems of the type

$$\text{find } x^* \in \underset{x \in C \cap L \cap Y}{\arg\min}\, c^T x \tag{P}$$

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

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



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

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

and C is a convex set.

THE MINLP PROBLEM SCOPE

The ESH algorithm solves convex MINLP problems of the type

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

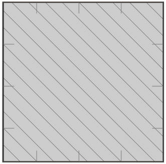
where $x = [x_1, x_2, \ldots, x_N]^T$ belongs to the compact set

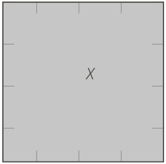
$$X = \{x \mid \underline{x}_i \leq x_i \leq \overline{x}_i,\ i = 1, \ldots, N\} \subset \mathbb{R}^n,$$

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

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

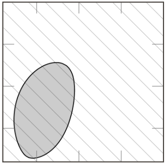
where $x = [x_1, x_2, \ldots, x_N]^T$ belongs to the compact set

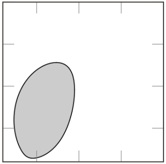
$$X = \{x \mid \underline{x}_i \leq x_i \leq \overline{x}_i,\ i = 1, \ldots, N\} \subset \mathbb{R}^n,$$

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

$$\text{find } x^* \in \underset{x \in C \cap L \cap Y}{\arg\min}\, c^T x \tag{P}$$

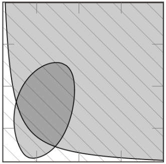
where $x = [x_1, x_2, \ldots, x_N]^T$ belongs to the compact set

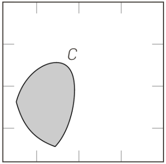
$$X = \{x \mid \underline{x}_i \leq x_i \leq \overline{x}_i,\ i = 1, \ldots, N\} \subset \mathbb{R}^n,$$

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.

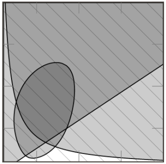




The ESH algorithm solves convex MINLP problems of the type

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

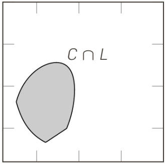
where $x = [x_1, x_2, \ldots, x_N]^T$ belongs to the compact set



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

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 ESH algorithm solves convex MINLP problems of the type

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

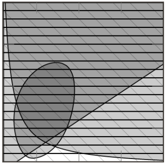
where $x = [x_1, x_2, \ldots, x_N]^T$ belongs to the compact set

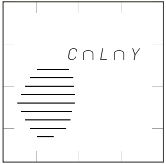
$$X = \{x \mid \underline{x}_i \leq x_i \leq \overline{x}_i,\ i = 1, \ldots, N\} \subset \mathbb{R}^n,$$

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 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 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

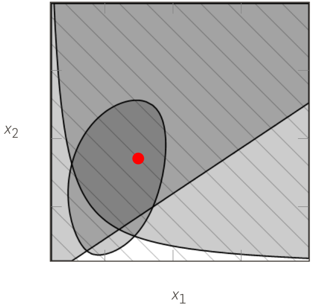
Obtain a feasible, relaxed interior point (satisfying C) by solving a NLP problem.

LP step (optional)

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

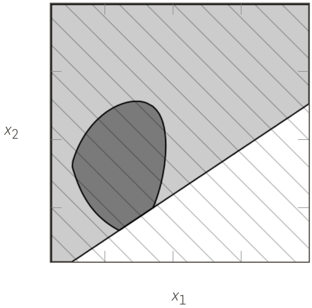
MILP step

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



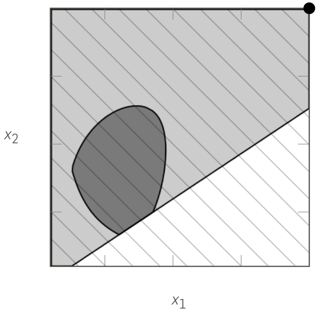
If an interior point is not given, obtain a feasible, relaxed interior point (satisfying all the nonlinear constraints in C) by solving a NLP problem.

LP STEP (OPTIONAL)

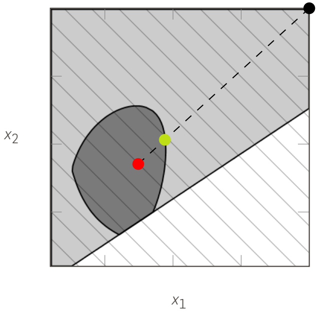


Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)

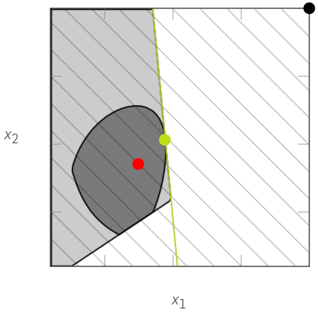


Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

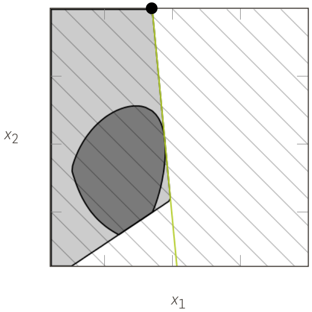


Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)

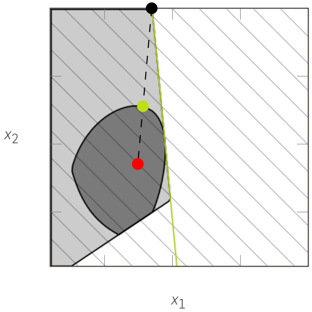


Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.



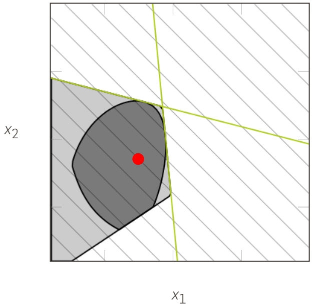
Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)



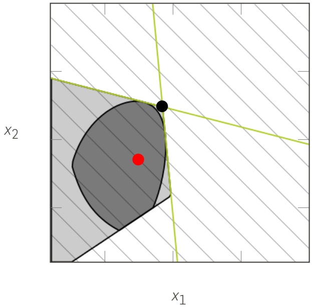
Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)



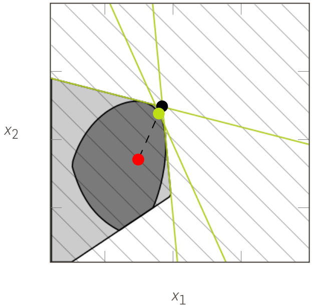
Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)

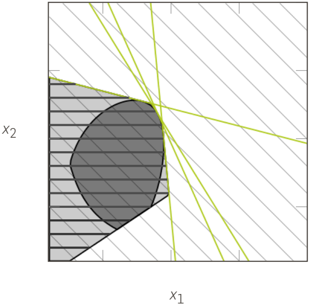


Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

LP STEP (OPTIONAL)

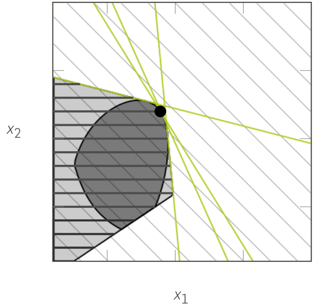


Solve simple LP problems and conduct a line search procedure to obtain supporting hyperplanes giving a first linear relaxation of the convex set C.

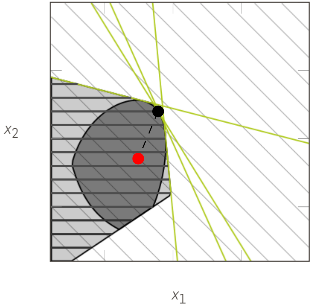


Finally include the integer requirements and solve MILP problems using a corresponding procedure to find the optimal solution to (P).

MILP STEP

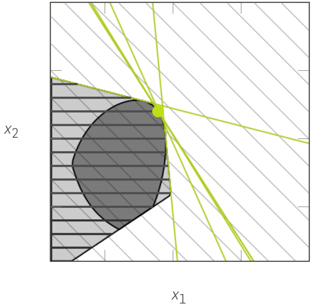


Finally include the integer requirements and solve MILP problems using a corresponding procedure to find the optimal solution to (P).



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

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

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

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

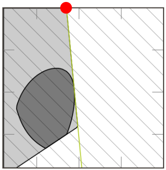
- utilizes several COIN-OR subprojects:
  Optimization Services (OS), Ipopt, CBC
- uses CPLEX, Gurobi, CBC for solving subproblems
- uses Ipopt for finding the interior point

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

DUAL AND PRIMAL BOUNDS

A **dual solution** provides a lower bound on the objective value:

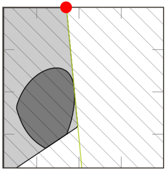
- Belongs to the relaxed set $\Omega_k \setminus (C \cap L)$.
- Provided by optimal solutions to the LP/MILP subproblems not in C.



A **dual solution** provides a lower bound on the objective value:

» Belongs to the relaxed set $\Omega_k \setminus (C \cap L)$.

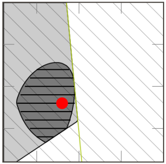
» Provided by optimal solutions to the LP/MILP subproblems not in C.



A **primal solution** provides an integer-feasible solution to the MINLP problem:

» Belongs to the feasible set $C \cap L \cap Y$.

» Provided by primal heuristics or the MILP solver.



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
- Can also be given directly as a bound reported by the MILP solver

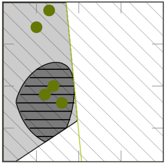
DUAL SOLUTIONS (LOWER BOUND)

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
- Can also be given directly as a bound reported by the MILP solver

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



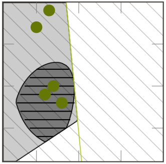
DUAL SOLUTIONS (LOWER BOUND)

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
- Can also be given directly as a bound reported by the MILP solver

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.

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

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

where DB and PB are the dual and primal objective values

Constraint feasibility tolerance

$$F(x^k_{\text{MILP}}) \leq \epsilon_{\text{MILP}}$$

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

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

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 $f(x) = x^T Qx + c^T x$, 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.

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 12.6.3 and IPOPT 3.11.7.

PERFORMANCE BENCHMARK

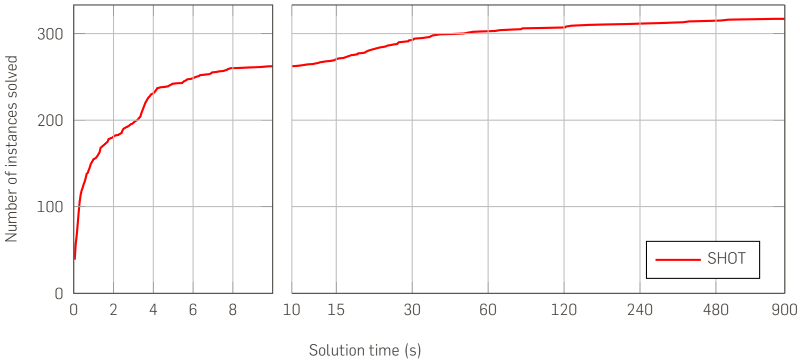
SHOT was tested on all 333 MINLP instances classified as convex in the MINLP Library 2:

- 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 12.6.3 and IPOPT 3.11.7.

Solution strategy

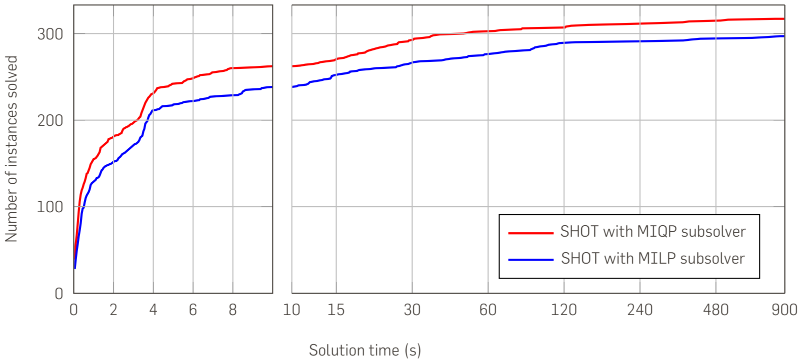
- $\epsilon_{\text{abs}} = \epsilon_{\text{rel}} = 0.001$, $\epsilon_{\text{MILP}} = 10^{-5}$, $\epsilon_{\text{LP}} = 0.001$, $K_{\text{LP}} = 300$
- maximal solution pool size: 10
- quadratic objective functions passed on to subsolver
- quadratic constraints regarded as general nonlinear

PERFORMANCE BENCHMARK



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.

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

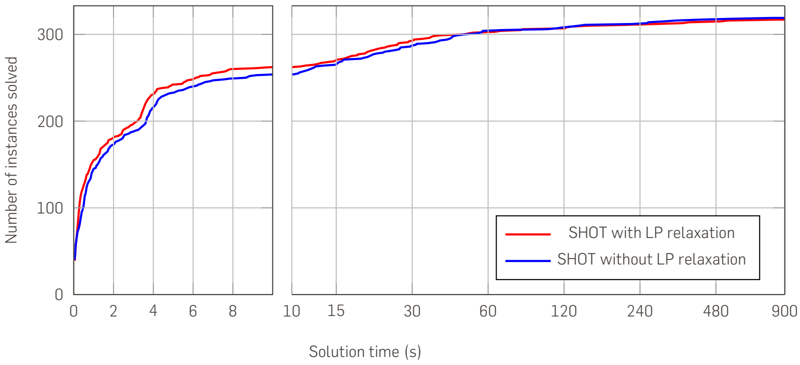
Initially integer-relaxed MILP/MIQP problems, *i.e.*, LP/QP problems, can be solved:

+ 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.

If the integer variable values are the same several subsequent iterations, these can be fixed and LP/QP problems solved instead of MILP/MIQP ones:

+ Much faster than fixing the variables and solving an NLP problem.
- Does not usually give a primal bound.

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.

UTILIZING LAZY CONSTRAINTS

In MILP solvers (CPLEX and Gurobi) so-called **lazy constraints** can be added “on the fly” through callbacks:

» A solution must fulfill the lazy constraints,
» but they are normally not included in the MILP model until needed.

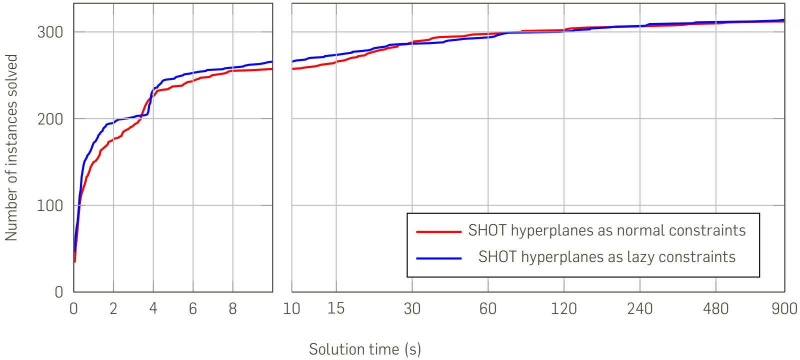
In SHOT early constraints can become redundant later:

» performance can be increased if these are not considered
» lazy constraints can also be purged if deemed unnecessary

When the MILP solver finds an integer-feasible solution, the callback is activated:

» creating a supporting hyperplane if the point is infeasible for the MINLP problem (considering tolerances)
» control is then returned to the MILP solver which can continue without rebuilding the branching tree.

LAZY CONSTRAINT STRATEGY IMPACT



Performance profiles of the number of problem instances solved by SHOT to an objective gap $\leq 1\%$ with the normal SHOT strategy and with the lazy constraint strategy.

BENCHMARKS AGAINST OTHER MINLP SOLVERS

BENCHMARKS AGAINST OTHER SOLVERS

The following MINLP solvers available in GAMS 24.4.1 were used:

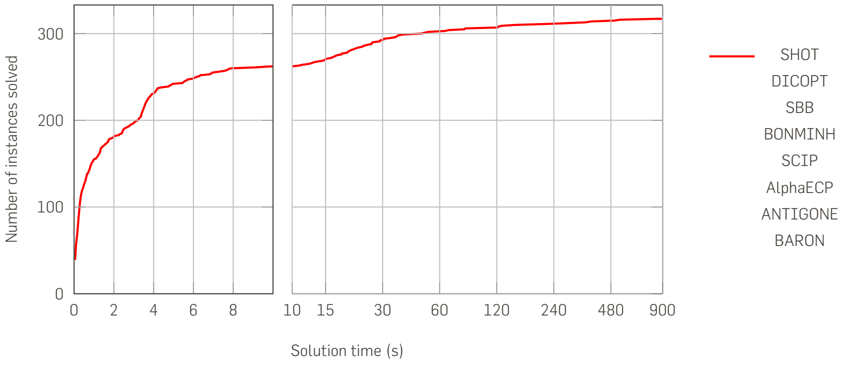
- AlphaECP (with convex strategy)
- ANTIGONE
- BARON
- BONMINH (with recommended convex strategy, B-Hyb)
- DICOPT
- SBB
- SCIP (with convex strategy)

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.

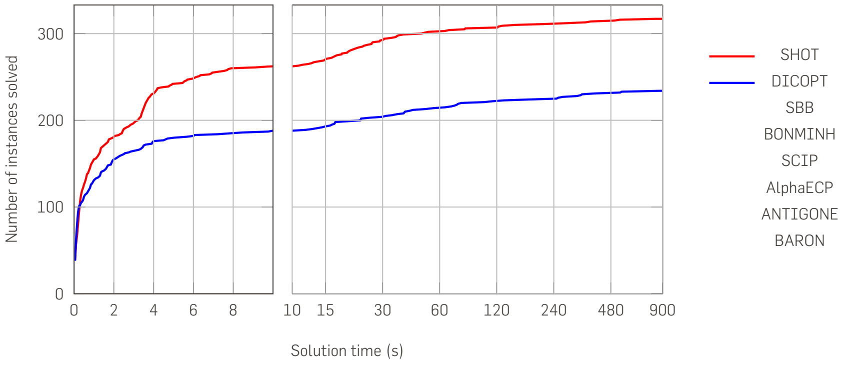
The total computational time for all solvers was about 150 h.

PERFORMANCE PROFILE



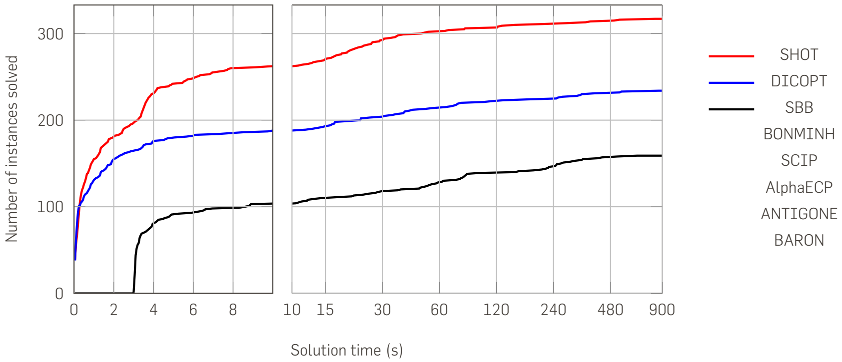
A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

PERFORMANCE PROFILE



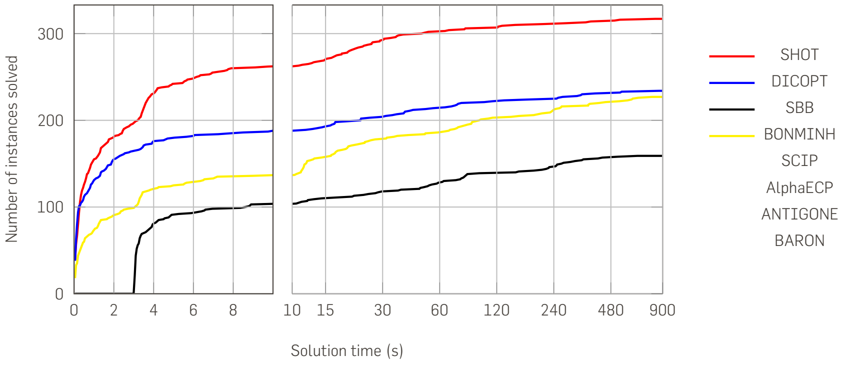
A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

PERFORMANCE PROFILE



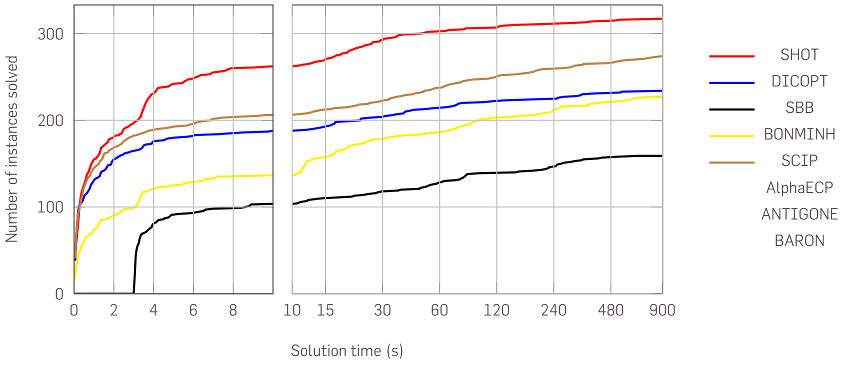
A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

PERFORMANCE PROFILE

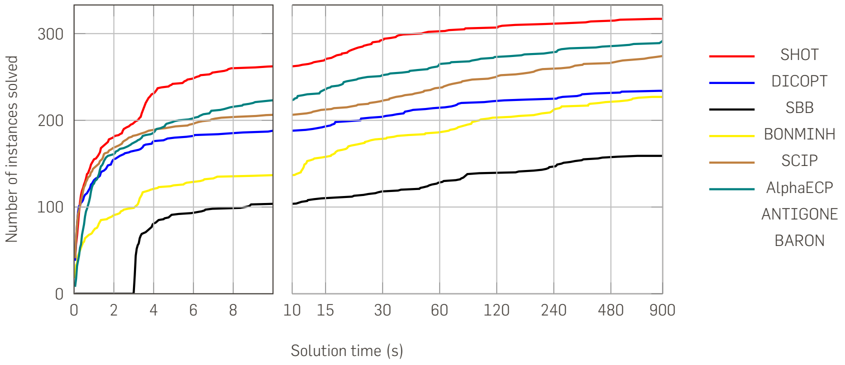


A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

PERFORMANCE PROFILE

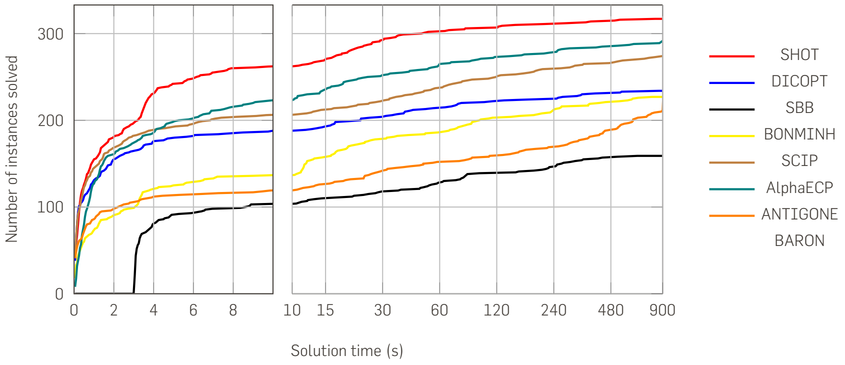


A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.



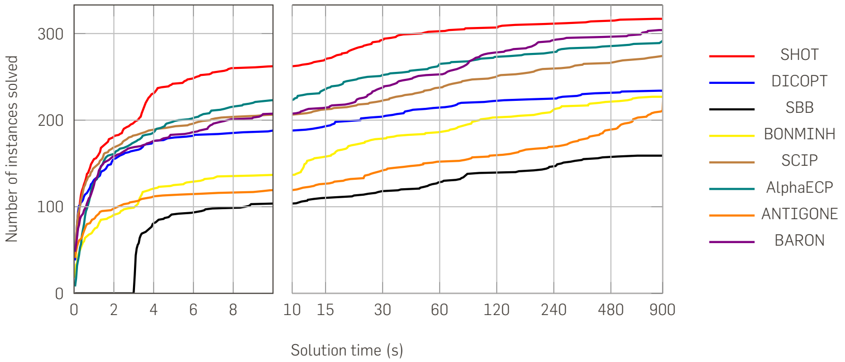
A performance profile of the number of problem instances solved to an objective duality gap $\leq 1\%$.

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.

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

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

CONCLUDING REMARKS

The ESH algorithm is a new method for convex MINLP.

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

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

Future research and development

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

CONCLUDING REMARKS

The ESH algorithm is a new method for convex MINLP.

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

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

Future research and development

» Investigate how to best select the interior point

» Improve handling of nonlinear objective functions

» Include the αR framework for global optimization of nonconvex MINLP problems

» Include automated reformulations for convex optimization

Thanks to GAMS for providing support for the benchmarks!

Thank you for your attention!

Thank you for your attention!

Questions or comments?