Artelys | OPTIMIZATION SOLUTIONS

Artelys Knitro 11.0

SOCP, API, Preconditioner and much more!

June 29, 2018

COMPANY OVERVIEW

Artelys

- We specialize in optimization, decision-support, modeling and deliver efficient solutions to complex business problems.

Domains of expertise

- Energy
- Transport & Logistic
- Defense
- Numerical and Combinatorial Optimization

Services

- Auditing & Consulting
- On demand software
- Distribution and support of numerical optimization tools
- Training

ArtelysTools

NUMERICAL OPTIMIZATION TOOLS

KNITRO

- Industry leading solver for very large, difficult nonlinear optimization problems (NLP, MINLP)

FICO Xpress Optimization Suite

- High performance linear, quadratic and mixed integer programming solver (LP,MIP,QP)

Artelys Kalis

- Object-oriented environment to model and solve problems with constraints programming techniques

AMPL

- Comprehensive modeling language for Mathematical Programming

Artelys | OPTIMIZATION SOLUTIONS

ARTELYS KNITRO: OVERVIEW

Background

- Created in 2001 by Ziena Optimization
  - Spin-off of Northwestern University
- Now developed and supported by Artelys

Key features

- Efficient and robust solution on **large scale** problems ($\sim 10^5$ variables)
- Four active-set and interior-point algorithms for continuous optimization
- MINLP algorithms and complementarity constraints for discrete optimization
- Many extra features based on **customer feedbacks** or project requirements
- Parallel multi-start method for global optimization.
- Easy to use and well documented: Online documentation

ARTELYS KNITRO: OVERVIEW

- Widely used in academia…
  - US Top Universities: Berkeley, Columbia, Harvard, MIT, Stanford…
  - Worldwide Top Universities: ETH Zürich, LSE, NUS (Singapore), Melbourne…

- … and industry
  - Economic consulting firms
  - Financial institutions
  - Mechanical engineering companies
  - Oil & Gas companies
  - Regulator & Policy maker
  - Software developers
    - ↳ Used as a third-party optimization engine

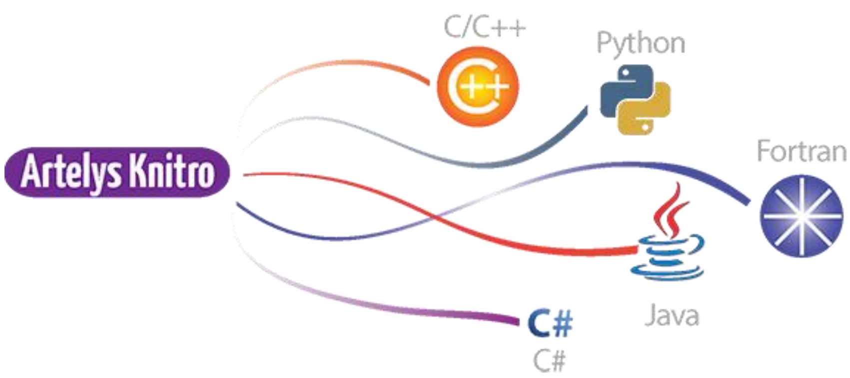
More than 400 institutions in over 40 countries rely on Artelys Knitro

Artelys | OPTIMIZATION SOLUTIONS

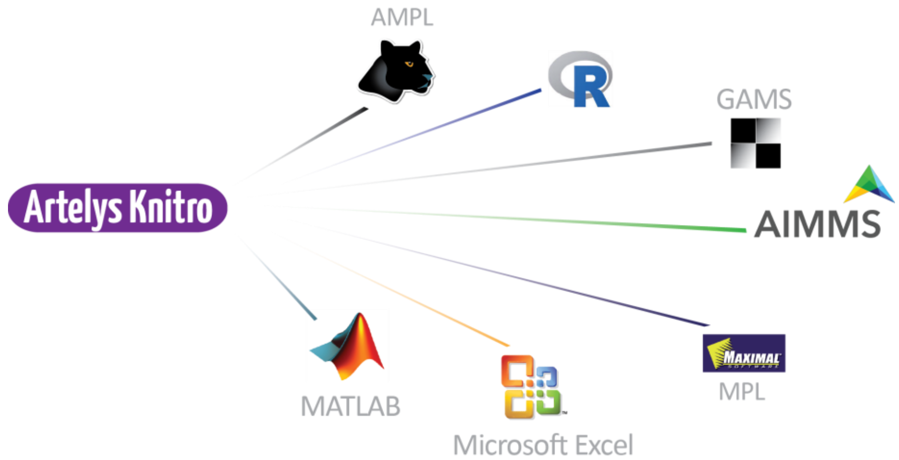
ARTELYS KNITRO: OVERVIEW

Interfaces

- Programming languages



- Modeling languages



Supported platforms

Windows

Windows 32-bit, 64-bit

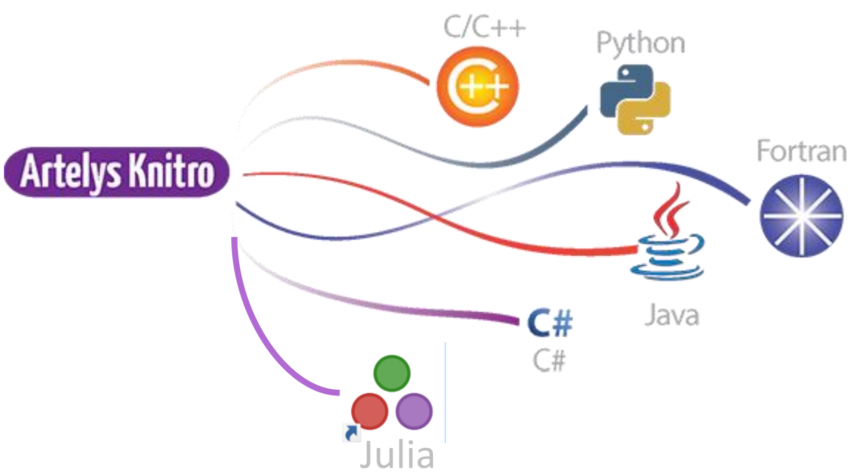
Linux 64-bit

macOS 64-bit

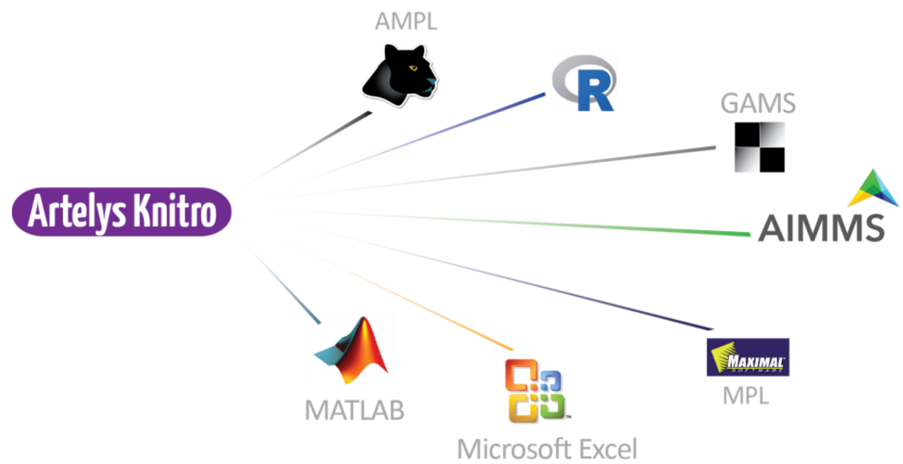
ARTELYS KNITRO: OVERVIEW

Interfaces

- Programming languages



- Modeling languages



Supported platforms

Windows 32-bit, 64-bit

Linux 64-bit

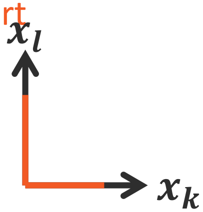
macOS 64-bit

Artelys | OPTIMIZATION SOLUTIONS

PROBLEM CLASSES SOLVED BY KNITRO

$$
\begin{array}{lll}
\min\limits_{x\in\mathbb{R}^n, y\in\mathbb{Z}^m} & f(x,y) & \\
\text{s.t.} & g_i(x,y) \geq 0 & i \in I \\
 & h_j(x,y) = 0 & j \in J \\
 & 0 \leq x_k \perp x_l \geq 0 & (k,l) \in C \subset [\![1,n]\!]^2 \\
 & l_x \leq x \leq u_x & \\
 & l_y \leq y \leq u_y &
\end{array}
$$

- Variables (x and y):
  - Continuous or discrete
  - Bounded or unbounded
- Objective (f) and constraints (g and h):
  - Linear or nonlinear
  - Smoothness: required, but may still work without it
  - Convexity: better but not required, local optimization or global optimization with multistart
- Complementarity constraints:
  - $0 \leq x_k \perp x_l \geq 0$ is equivalent to: $x_k \geq 0$ and $x_l \geq 0$ and $\{x_k = 0 \text{ or } x_l = 0\}$
  - Applications: strategic bidding, economic models, equilibrium constraints, disjunctions



Artelys Knitro 11.0 new features:

- New SOCP Algorithm
  - Detect conic constraints from quadratic structures
  - Designed for **general nonlinear problems with SOC constraints**
- New C API
  - Easier to use
  - Allows passing problem structure (eg linear, quadratic, **conic constraints**) with dedicated API and without providing Hessian
- Preconditioning for all classes of problems
  - **Preconditioning** can now be used for problems with **equality** and inequality constraints
- New parallel linear solvers
  - HSL MA86 (non-deterministic) and MA97 (deterministic)
  - Speedups on large scale problem with shared memory parallelism
- Performance improvements
  - Very large speedup on QCQPs (including nonlinear QCQP)
  - Speedups on general convex problems
  - Speedups on MINLP algorithms

Artelys Knitro 11.0 new features:

1
- New SOCP Algorithm
  - Detect conic constraints from quadratic structures
  - Designed for **general nonlinear problems with SOC constraints**
- New C API
  - Easier to use
  - Allows passing problem structure (eg linear, quadratic, **conic constraints**) with dedicated API and without providing Hessian
- Preconditioning for all classes of problems
  - **Preconditioning** can now be used for problems with **equality** and inequality constraints
- New parallel linear solvers
  - HSL MA86 (non-deterministic) and MA97 (deterministic)
  - Speedups on large scale problem with shared memory parallelism
- Performance improvements
  - Very large speedup on QCQPs (including nonlinear QCQP)
  - Speedups on general convex problems
  - Speedups on MINLP algorithms

Artelys Knitro 11.0 new features:

- New SOCP Algorithm
  - Detect conic constraints from quadratic structures
  - Designed for **general nonlinear problems with SOC constraints**
- New C API
  - Easier to use
  - Allows passing problem structure (eg linear, quadratic, **conic constraints**) with dedicated API and without providing Hessian

2

- Preconditioning for all classes of problems
  - **Preconditioning** can now be used for problems with **equality** and inequality constraints
- New parallel linear solvers
  - HSL MA86 (non-deterministic) and MA97 (deterministic)
  - Speedups on large scale problem with shared memory parallelism
- Performance improvements
  - Very large speedup on QCQPs (including nonlinear QCQP)
  - Speedups on general convex problems
  - Speedups on MINLP algorithms

Artelys Knitro 11.0 new features:

- New SOCP Algorithm
  - Detect conic constraints from quadratic structures
  - Designed for **general nonlinear problems with SOC constraints**
- New C API
  - Easier to use
  - Allows passing problem structure (eg linear, quadratic, **conic constraints**) with dedicated API and without providing Hessian
- 3 Preconditioning for all classes of problems
  - **Preconditioning** can now be used for problems with **equality** and inequality constraints
- New parallel linear solvers
  - HSL MA86 (non-deterministic) and MA97 (deterministic)
  - Speedups on large scale problem with shared memory parallelism
- Performance improvements
  - Very large speedup on QCQPs (including nonlinear QCQP)
  - Speedups on general convex problems
  - Speedups on MINLP algorithms

SOCP

- Standard Second Order Cone (SOC) of dimension *k* is

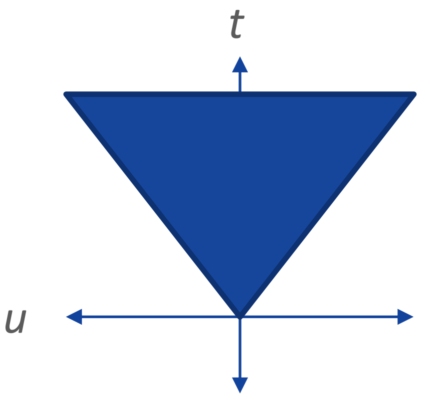
$$\left\{ \begin{bmatrix} t \\ u \end{bmatrix} \middle| u \in R^{k-1}, t \in R^1, \|u\| \leq t \right\}$$

- For *k=1*

$$\{t |\ t \in R^1, 0 \leq t\}$$

- For *k=2*

$$\left\{ \begin{bmatrix} t \\ u \end{bmatrix} \middle| u \in R^1, t \in R^1, |u| \leq t \right\}$$



SECOND ORDER CONE PROGRAMS (SOCP)

- Set $u = Ax + b$ and $t = c^{\mathrm{T}}x + d$ to create general second order cone constraints of form

$$\|Ax + b\| \leq c^T x + d$$

- Second Order Cone Program (SOCP):

$$\min_x f^{\mathrm{T}} x$$

$$\text{s.t.} \qquad \|A_i x + b_i\| \leq c_i^{\mathrm{T}} x + d_i, \quad i\text{=1..m}$$

$$G^T x + h \leq 0$$

  - Convex QP and QCQP (and more) can be converted to SOCP

APPLICATIONS

- Applications
  - Finance: portfolio optimization with loss risk constraints
  - Facility location (e.g. antenna placement in wireless network)
  - Robust optimization (under ellipsoid uncertainty)
  - Robust least squares
  - Grasping force optimization
  - FIR filter design
  - Truss design

- See *Applications of Second-Order Cone Programming,* Lobo,Vandenberghe, Boyd, Lebret

EXAMPLES

- Quadratic constraint

$$x^T Qx + 2q^T x + r \leq 0$$
$$\left\|Q^{1/2}x + Q^{-1/2}q\right\|^2 + r - q^T Q^{-1}q \leq 0$$
$$||Q^{1/2}x + Q^{-1/2}q|| \leq (q^T Q^{-1}q - r)^{1/2}$$

- Rotated cone constraint

$$x^T x \leq yz, y \geq 0, z \geq 0$$
$$4x^T x + y^2 + z^2 \leq 4yz + y^2 + z^2$$
$$\sqrt{4x^T x + (y - z)^2} \leq y + z$$
$$\left\|\begin{matrix} 2x \\ y - z \end{matrix}\right\| \leq y + z$$

KNITRO CONIC SOLVER

- Knitro identifies the constraints in form

$$\sum_{i=1}^{n} a_i x_i^2 \leq a_0 x_0^2, \quad x_0 \geq 0$$

and

$$\sum_{i=2}^{n} a_i x_i^2 \leq a_0 x_0 x_1, \quad x_0, x_1 \geq 0.$$

as second order cone constraints, and internally puts them into the standard form

- It also allows to input constraints in form

$$||Ax + b|| \leq cx + d$$

directly via the struct ‘L2norm’

- Currently, it does second order conic constraint identification on the presolved problem

KNITRO CONIC SOLVER

- Knitro conic solver moves beyond SOCP (more general)

$$\min_x f^{\mathrm{T}} x$$
$$\text{s.t.} \quad \|A_i x + b_i\| \leq c_i^{\mathrm{T}} x + d_i$$
$$G^T x + h \leq 0$$

$$\min_x f(x)$$
$$\text{s.t.} \quad \|A_i x + b_i\| \leq c_i^{\mathrm{T}} x + d_i$$
$$h(x) = 0$$
$$g(x) \leq 0$$

- Handle any nonlinear problem with second order cone constraints, including non-convex
- First specialized solver of this kind !
- Extension of existing Knitro Interior/Direct algorithm

KNITRO CONIC SOLVER

- It generalizes the operations on the slack variables in the existing Knitro/Direct algorithm using the algebra associated with second order cones

$$\begin{aligned} \min \quad & f(x) \\ & g(x) \le 0 \\ & H(x) \in K. \end{aligned}$$

$$H(x) := Mx + v = \begin{pmatrix} c_1 \\ A_1 \\ \dots \\ c_t \\ A_t \end{pmatrix} x + \begin{pmatrix} d_1 \\ b_1 \\ \dots \\ d_t \\ b_t \end{pmatrix}$$

⇓

$$\begin{aligned} \min \quad & f(x) \\ & s := -g(x), \; s \ge 0 \\ & y := H(x), \; y \in K. \end{aligned}$$

$$\nabla f(x) + A^T \lambda - M^T w = 0$$

$$g(x) \le 0, \lambda \ge 0$$
$$H(x) \in K, w \in K$$

$$g(x).\lambda = 0$$
$$H(x) \circ w = 0$$

NLP FORM VS. CONIC FORM

- Can always write the cone constraint as a general NLP constraint:

$$||u|| \leq t \;\rightarrow\; \sqrt{u_1^2 + u_2^2 + .. + u_{k-1}^2} \leq t$$

- This does not work well in general; constraint is non-differentiable as $||u|| \rightarrow 0$
  - It is not uncommon that $||u|| \rightarrow 0$ at the optimal solution

- Can square the constraint, but then it is non-convex and degenerate at the solution if $||u|| \rightarrow 0$
- Can try to smooth or relax/perturb these constraints to avoid these issues, e.g.

$$\sqrt{u_1^2 + u_2^2 + .. + u_{k-1}^2 + \epsilon^2} \leq t$$

  - This works better sometimes but is still not robust or nearly as effective as dealing with them directly

NLP FORM VS. CONIC FORM

- Consider the simple example:

$$\min_x 0.5x_1 + x_2$$
$$\text{s.t.} \quad |x_1| \le x_2$$

  - The optimal solution is at (0,0)
  - NLP form without conic detection can't get dual feasible
  - QCQP form (non-convex):

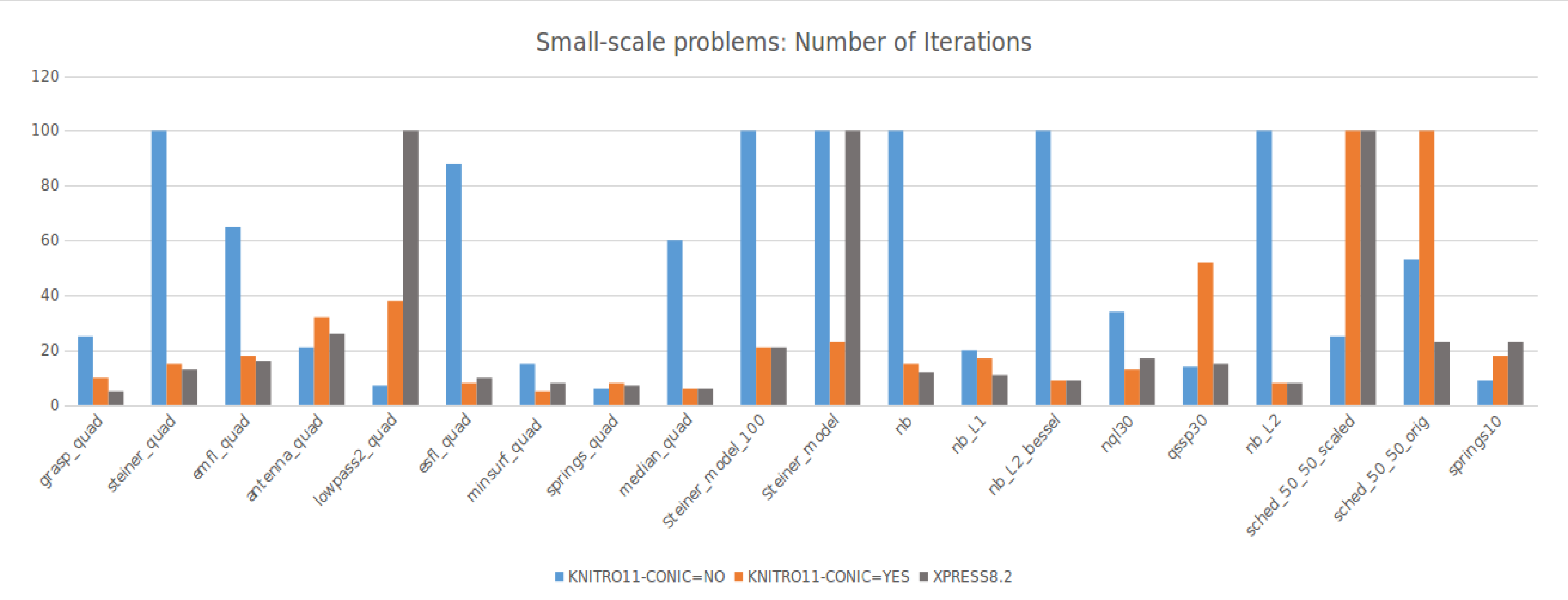
$$\min_x 0.5x_1 + x_2$$
$$\text{s.t.} \quad x_1^2 \le x_2^2, \ x_2 \ge 0$$

    solves in 50 iterations
  - Conic formulation solves in 4 iterations

- In fact, development of the conic extension started after having troubles with such a problem
  - Steiner_model / Steiner_model_100 on the next slide

NLP FORM VS. CONIC FORM

Compare Knitro (with and without special treatment of cone constraints) and Xpress on small SOCP models (iteration comparison)



KNITRO CONIC SOLVER-MISOCP

- Can utilize Knitro branch-and-bound algorithm to solve mixed-integer SOCP

- … or more general mixed-integer models with SOC constraints

$$\min_{x,y} f(x,y)$$

$$\text{s.t. } \|A_i x + b_i\| \le c_i^T x + d_i, \quad i=1..k$$

$$h(x,y) = 0$$

$$g(x,y) \le 0$$

$$y \text{ integer}$$

NEW KNITRO 11.0 API

NEW KNITRO 11.0 API – KEY FEATURES

- Build optimization model piece-by-piece
  - More flexible
  - Easier problem modification
  - More extendable (to multi-objective, statistical learning models, etc.)
- Identify special structures (e.g. linear, quadratic, conic, etc)
  - Identify more problem types (QCQP, SOCP, etc)
  - Potential for more extensive presolve operations
  - Faster (potentially parallel) evaluations of stored structures
- Can combine exact and approximate derivatives

C EXAMPLE

$$\min f(x) + (1 - x_0)^2$$

$$\text{s.t.} \quad x_0 x_1 \geq 1$$

$$x_0 + x_1^2 \geq 0, \qquad x_0 \leq 0.5$$

C EXAMPLE

$$\min f(x) + (1 - x_0)^2$$

$$\text{s.t.} \quad x_0 x_1 \geq 1$$

$$x_0 + x_1^2 \geq 0, \qquad x_0 \leq 0.5$$

```c
// Create a new Knitro solver instance.
KN_new(&kc);

// Add variables and constraints and set their bounds
KN_add_vars(kc, 2, NULL);
KN_set_var_upbnd(kc, 0, 0.5);

KN_add_cons(kc, 2, NULL);
double cLoBnds[2] = {1.0, 0.0};
KN_set_con_lobnds_all(kc, cLoBnds);
```

C EXAMPLE

$$\min f(x) + (1 - x_0)^2$$

s.t. $x_0 x_1 \geq 1$

$x_0 + x_1^2 \geq 0, \qquad x_0 \leq 0.5$

```
// Load quadratic structure x0*x1 for the first constraint
indexVar1 = 0; indexVar2 = 1; coef = 1.0;
KN_add_con_quadratic_struct_one (kc, 1, 0,
                                 &indexVar1, &indexVar2, &coef);
```

C EXAMPLE

$$\min f(x) + (1 - x_0)^2$$

$$\text{s.t.} \quad x_0 x_1 \geq 1$$

$$x_0 + x_1^2 \geq 0, \qquad x_0 \leq 0.5$$

```
// Add linear term x0 in the second constraint
indexVar = 0; coef = 1.0;
KN_add_con_linear_struct_one (kc, 1, 1,
                              &indexVar, &coef);

// Add quadratic term x1^2 in the second constraint
indexVar1 = 1; indexVar2 = 1; coef = 1.0;
KN_add_con_quadratic_struct_one (kc, 1, 1,
                              &indexVar1, &indexVar2, &coef);
```

C EXAMPLE

$$\min f(x) + (1 - x_0)^2$$

$$\text{s.t.} \quad x_0 x_1 \geq 1$$

$$x_0 + x_1^2 \geq 0, \qquad x_0 \leq 0.5$$

```
// Pointer to structure holding information for callback
CB_context   *cb;

// Add a callback function "callbackEvalF" to evaluate the nonlinear
//  (non-quadratic) part of the objective
KN_add_eval_callback (kc, KNTRUE, 0, NULL,  f(x), &cb);

// Add the constant, linear, and quadratic terms in the objective.
KN_add_obj_constant(kc, 1.0);
indexVar = 0; coef = -2.0;
KN_add_obj_linear_struct(kc, 1, &indexVar, &coef);
indexVar1 = 1; indexVar2 = 1; coef = 1.0;
KN_add_obj_quadratic_struct(kc, 1, &indevVar1, &indexVar2, &coef);
```

C EXAMPLE

$$\min f(x) + (1 - x_0)^2$$

$$\text{s.t.} \quad x_0 x_1 \geq 1$$

$$x_0 + x_1^2 \geq 0, \qquad x_0 \leq 0.5$$

```
// Set the non-default SQP algorithm
KN_set_int_param(kc,KN_PARAM_ALGORITHM, KN_ALG_ACT_SQP);

// Solve the problem.
KN_solve (kc);

// An example of obtaining solution information.
KN_get_solution(kc, &nStatus, &objSol, x, lambda);

// Delete the Knitro solver instance.
KN_free (&kc);
```

NEW KNITRO 11.0 API – PERFORMANCE

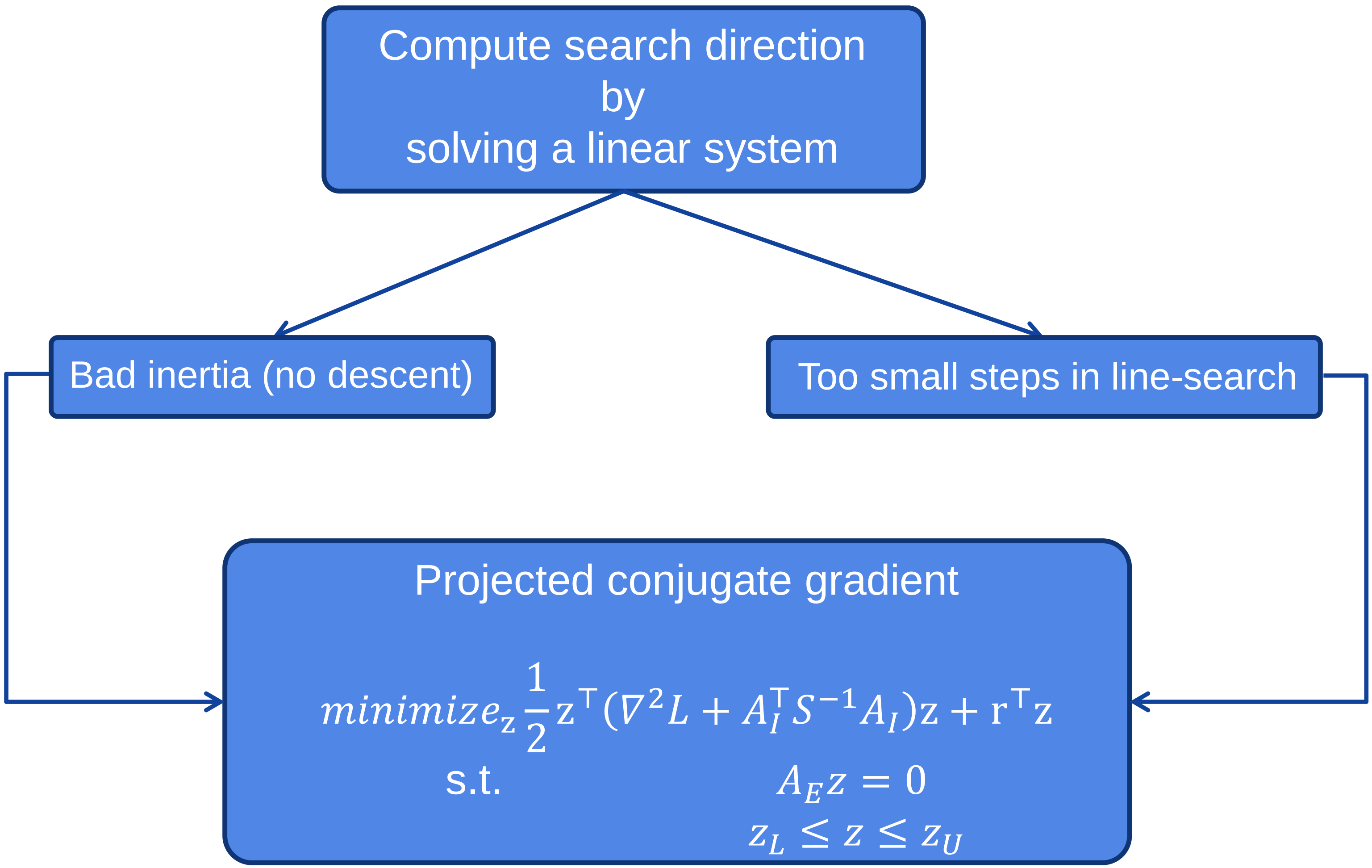
- Comparing old API and new API on some large QCQP models

Problem	#nnzJ	#nnzH	Old API (solve time)	New API (solve time)
qcqp1000-1nc	5,591	83,872	33.27	**27.41**
qcqp1000-2c	63,139	142,386	20.19	**9.20**
qcqp1000-2nc	63,139	131,114	17.71	**8.38**
qcqp1500-1c	180,041	438,989	1322.30	**393.09**
qcqp1500-1nc	180,041	409,820	**230.93**	330.52
qcqp500-3c	5,685	125,086	16.30	**0.71**
qcqp500-3nc	5,686	125,086	17.79	**0.72**
qcqp750-2c	10,792	281,514	56.37	**2.21**
qcqp750-2nc	10,792	281,514	55.37	**2.25**

PRECONDITIONER

PRECONDITIONING IN KNITRO

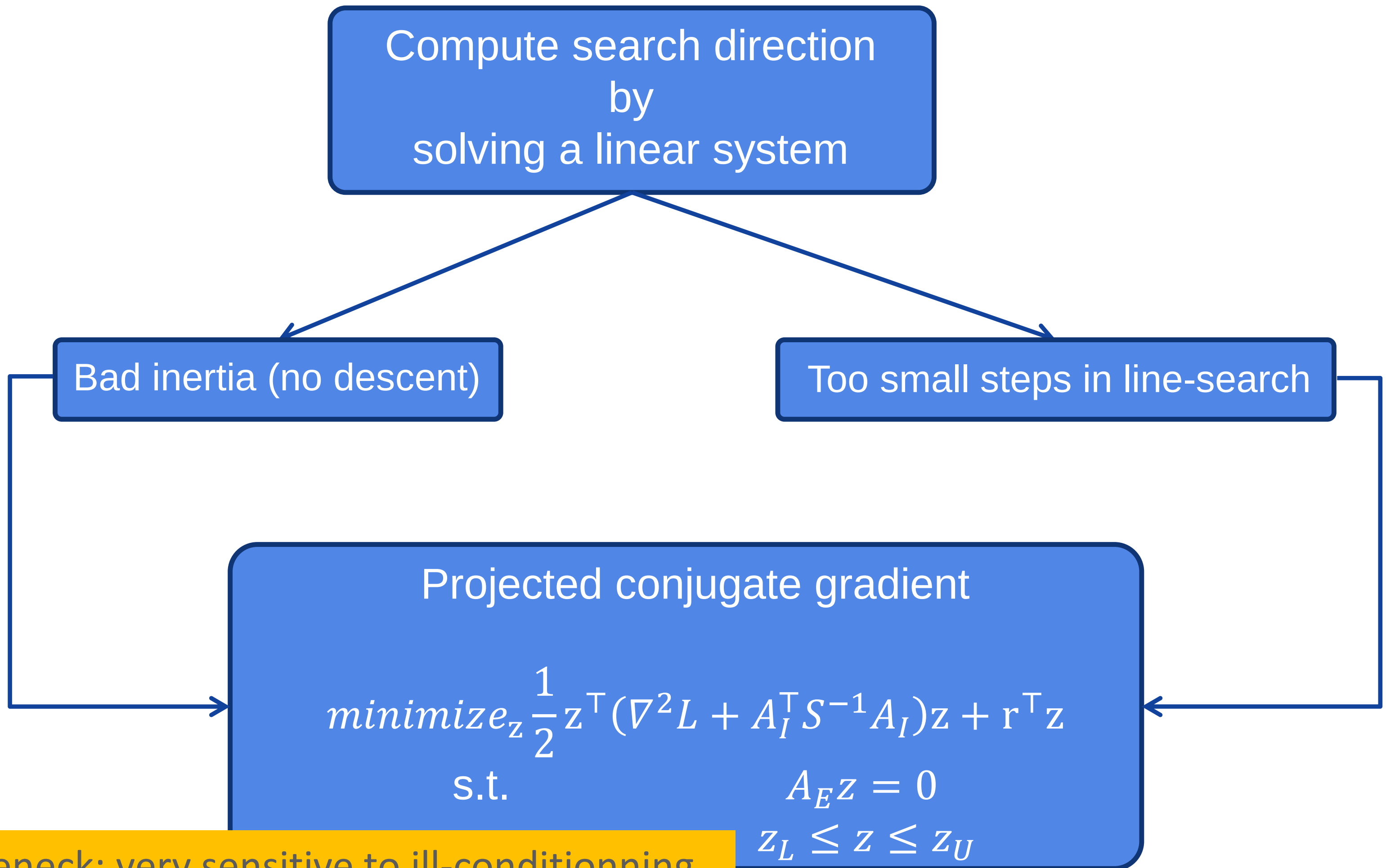
- Fallback step in projected conjugate gradient (PCG) with Knitro's interior point



Artelys OPTIMIZATION SOLUTIONS

PRECONDITIONING IN KNITRO

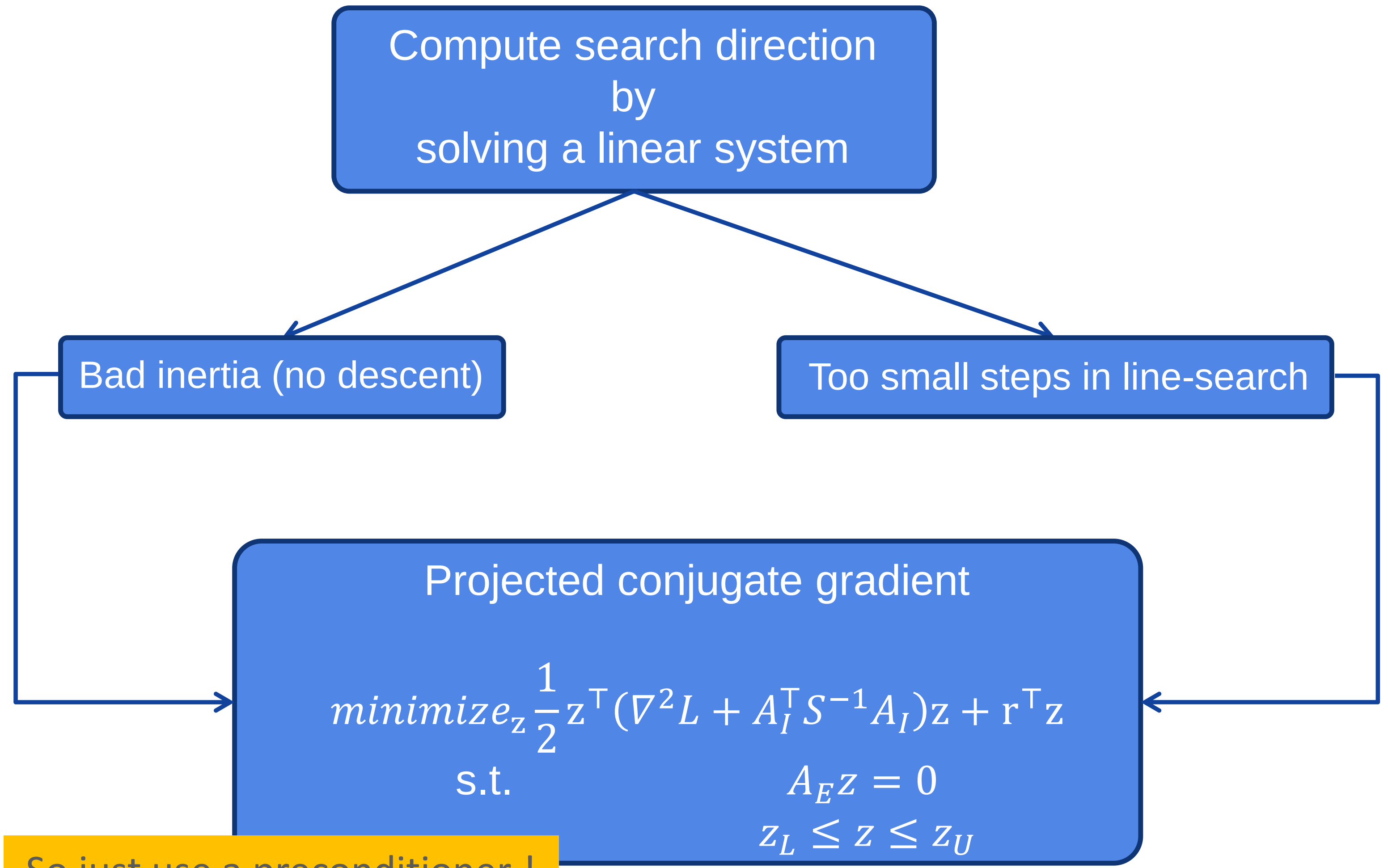
- Fallback step in projected conjugate gradient (PCG) with Knitro’s interior point



Bottleneck: very sensitive to ill-conditionning

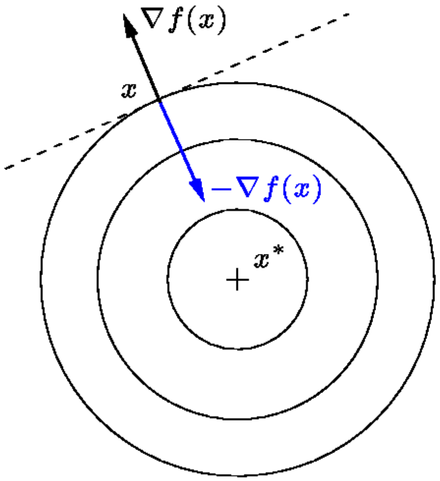
PRECONDITIONING IN KNITRO

- Fallback step in projected conjugate gradient (PCG) with Knitro's interior point

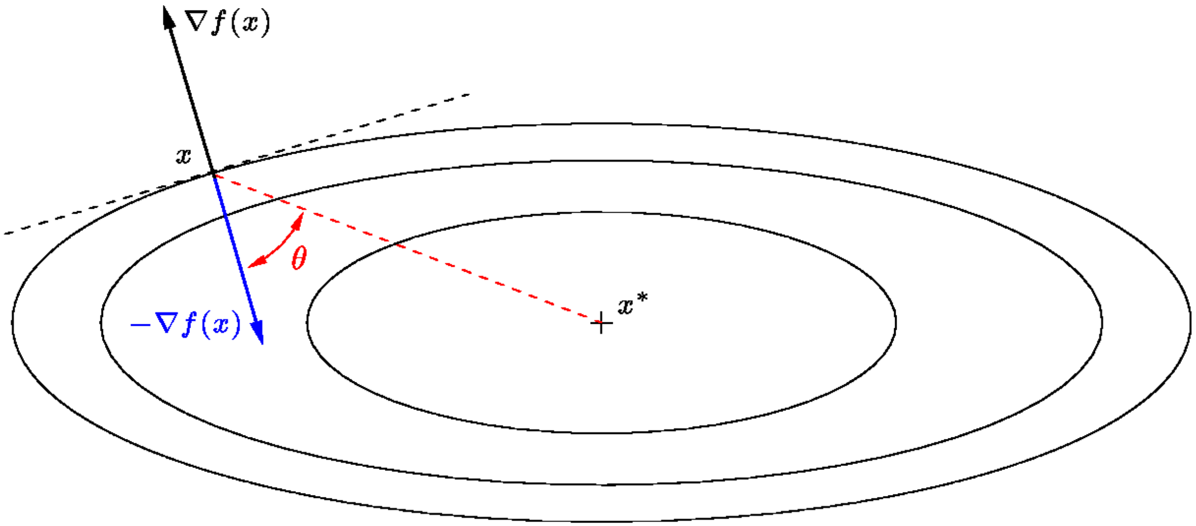


So just use a preconditioner !

CONDITIONING

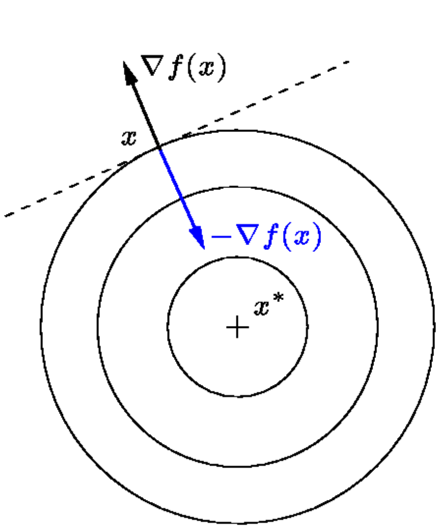


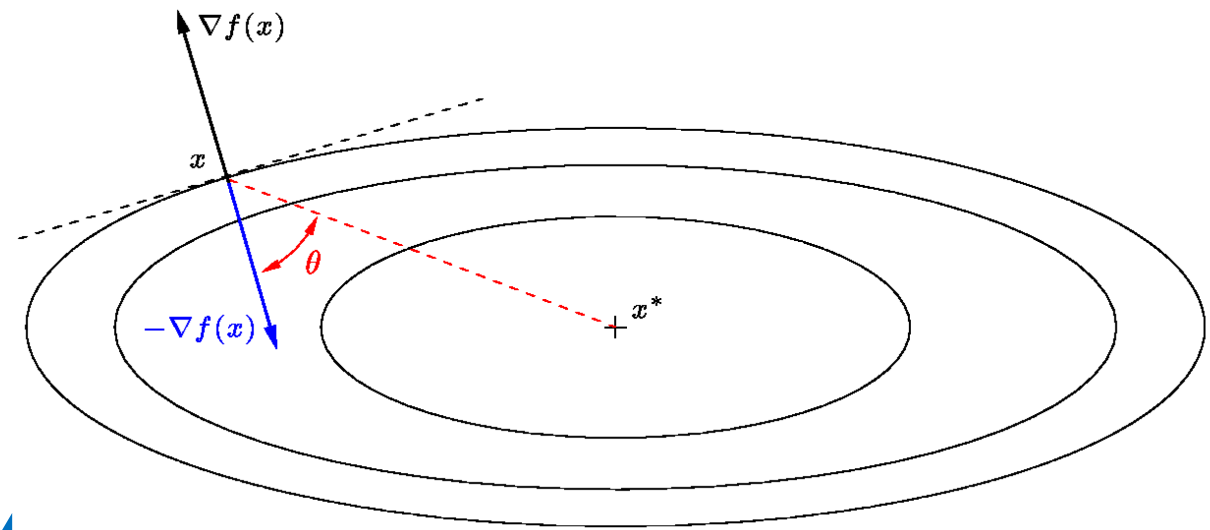
Good conditioning



Bad conditioning

CONDITIONING





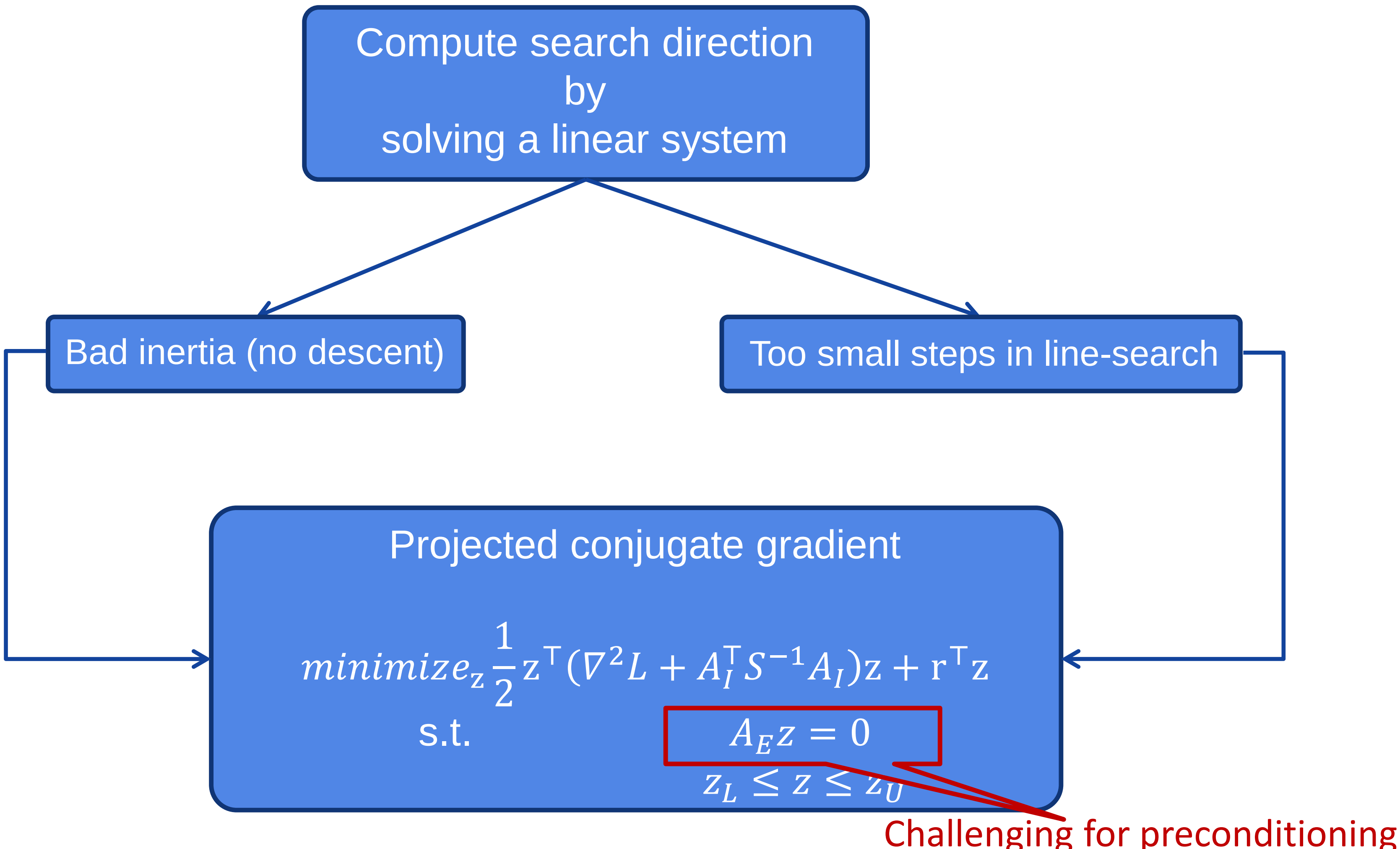
Good conditioning ← **Preconditioner** ← Bad conditioning

y = Mx

Artelys | OPTIMIZATION SOLUTIONS

PRECONDITIONING IN KNITRO

- Fallback step in projected conjugate gradient (PCG) with Knitro’s interior point



PRECONDITIONER'S MECHANICS

- **Incomplete Choleski factorization** (*icfs* module)

$$\nabla_{xx}^2 L + A_I^\top S^{-1} \Lambda A_I \approx LL^T$$

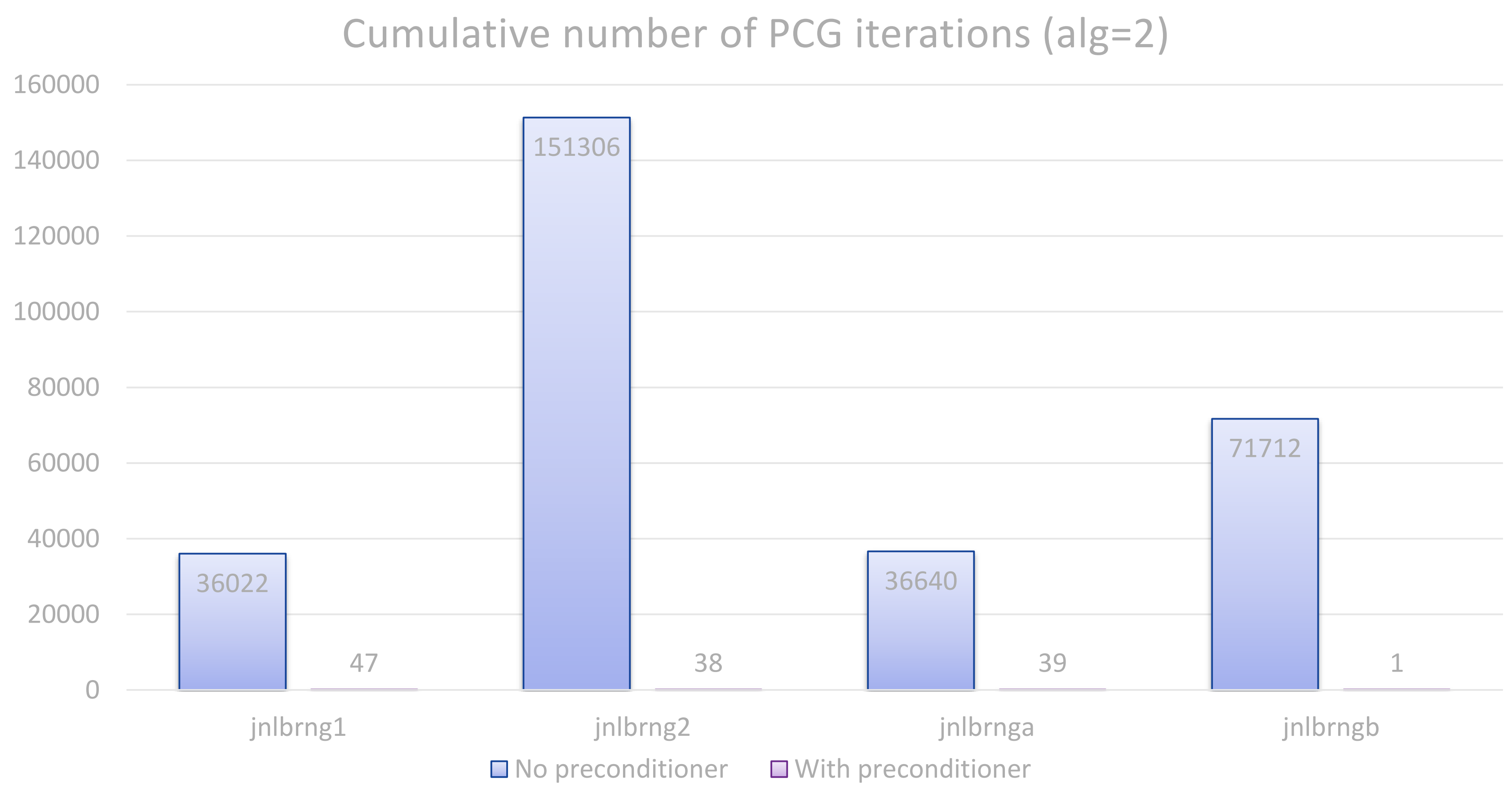
Steps to transform PCG direction so that $\boldsymbol{A_E z = 0}$

1) Compute $\tilde{r} := L^{-1} r$
2) Form the dense matrix $B := L^{-1} \boldsymbol{A}_{\boldsymbol{E}}^\top$
3) Compute $C := B^\top B$
4) Solve Cw = $B^\top \tilde{r}$
5) Compute $\tilde{z} = \tilde{r} - Bw$
6) Backsolve $z = L^{-\top} \tilde{z}$

New Knitro options
cg_precond (0 or 1)
cg_pmem (density of the incomplete Cholesky factorization)

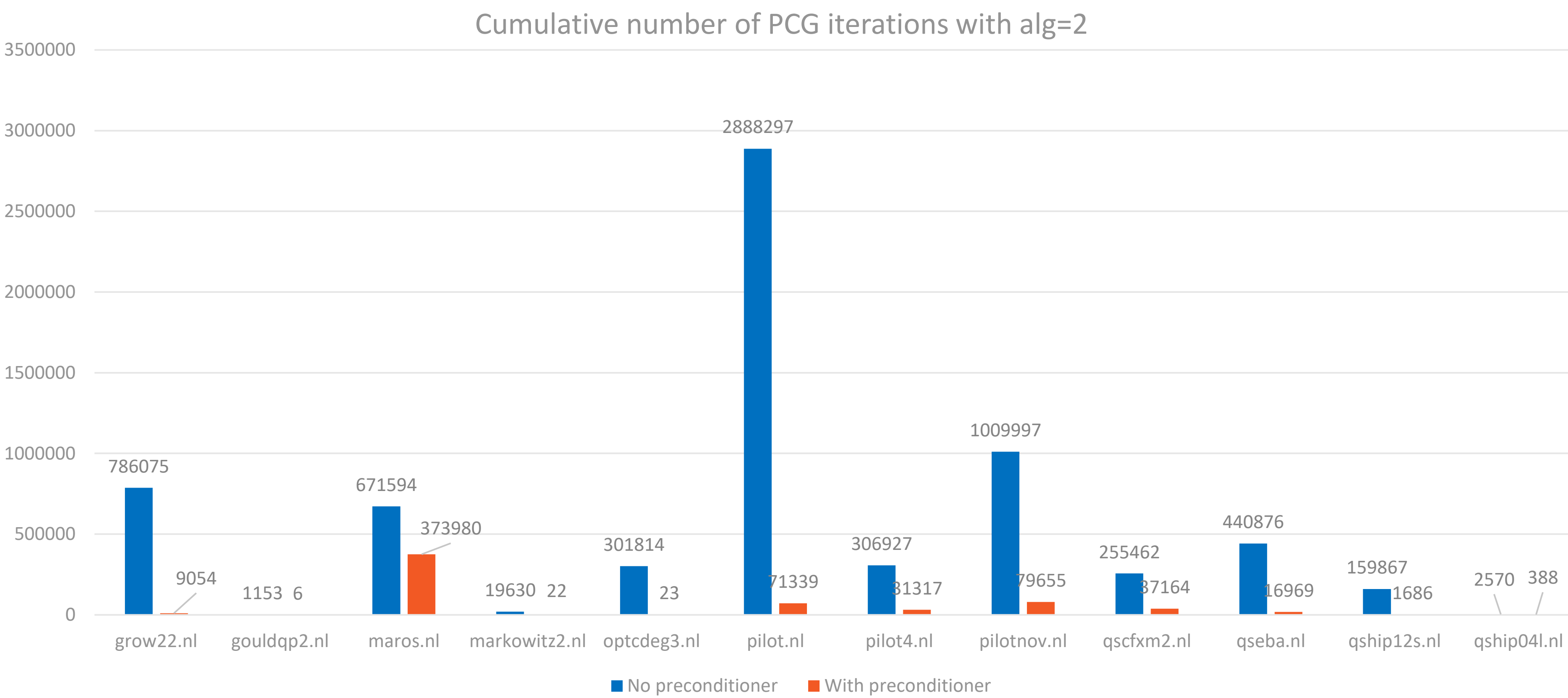
PERFORMANCE IMPROVEMENTS

- Nonlinear programs with inequality constraints only (alg=2, Knitro PCG)



PERFORMANCE IMPROVEMENTS

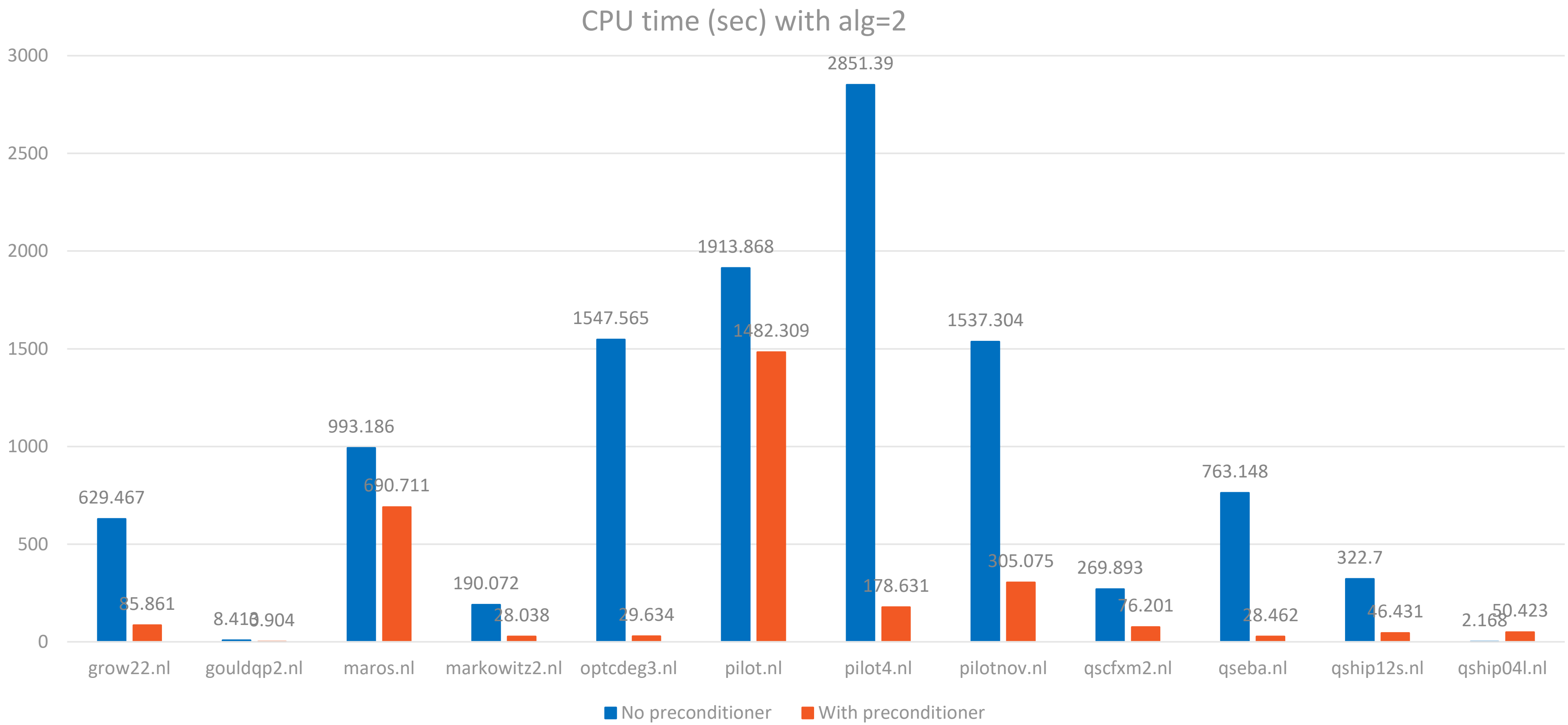
- Nonlinear programs with equality constraints (alg=2, Knitro PCG)



# eqs	440	349	265	402	800	208	279	632	352	441	316	277
# ineqs	880	699	419	201	800	1176	123	233	210	1	101	40

PERFORMANCE IMPROVEMENTS

- Nonlinear programs with equality constraints (alg=2, Knitro PCG)

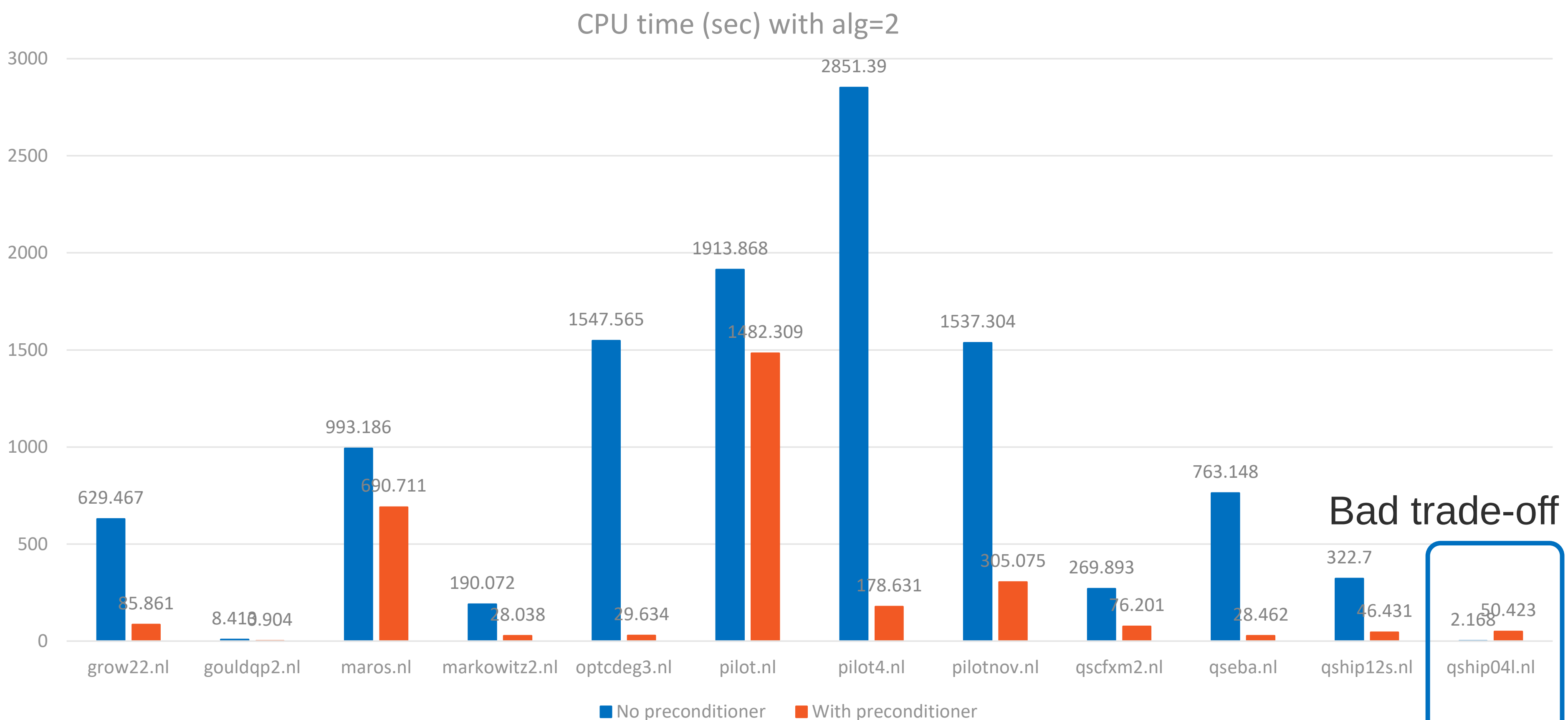


# eqs	440	349	265	402	800	208	279	632	352	441	316	277
# ineqs	880	699	419	201	800	1176	123	233	210	1	101	40

Artelys | OPTIMIZATION SOLUTIONS

PERFORMANCE IMPROVEMENTS

- Nonlinear programs with equality constraints (alg=2, Knitro PCG)



# eqs	440	349	265	402	800	208	279	632	352	441	316	277
# ineqs	880	699	419	201	800	1176	123	233	210	1	101	40

LINEAR SOLVERS

LINEAR SOLVER OPTIONS IN ARTELYS KNITRO

- Knitro algorithms need to solve sparse symmetric indefinite linear systems, Ax=b, where A and b depends on the iteration
  - Although the factorization is unique, each solver uses a different algorithm to compute it (multifrontal/supernodal; pivoting; etc.)
  - The choice of the linear solver can change the number of iterations taken to solve a problem, and sometimes even the return status
  - There is no linear solver best for all problems

- Artelys Knitro 11.0 allows the use of two more parallel linear solvers
  - HSL MA27 sequential
  - HSL MA57 sequential
  - MKL PARDISO parallel
  - HSL MA86 parallel
  - HSL MA97 parallel; bit-compatible (always give the same answer)

LINEAR SOLVER OPTIONS IN ARTELYS KNITRO

- MA57 performs well for small and medium size problems
- …but it might have no chance in the large scale
  - Problem MSK_STEP3

Number of variables	55,163
Number of constraints	108,911
Number of nonzeros in Jacobian	54,330,672
Number of nonzeros in Hessian	3,361,333

  - MA57 : out of memory

	4 threads	8 threads	16 threads	30 threads
MKLPARDISO				
# of iterations	90	90	90	90
Total program time (secs)	4704.27100 (14239.626 CPU time)	3646.25830 (18465.613 CPU time)	3158.46362 (27331.607 CPU time)	3156.07690 (49937.773 CPU time)
KKT Factorization time/count	3168.19702 / 96	2104.03540 / 96	1592.80664 / 96	1589.04285 / 96
MA86				
# of iterations	90	90	90	90
Total program time (secs)	3363.03809 (11535.405 CPU time)	2370.03833 (14391.964 CPU time)	2111.79712 (23614.680 CPU time)	2265.83179 (47518.672 CPU time)
KKT Factorization time/count	2689.55811 / 90	1726.85120 / 90	1461.42456 / 90	1609.26416 / 90

Artelys | OPTIMIZATION SOLUTIONS

Your turn!

Try it and let us know what you think…

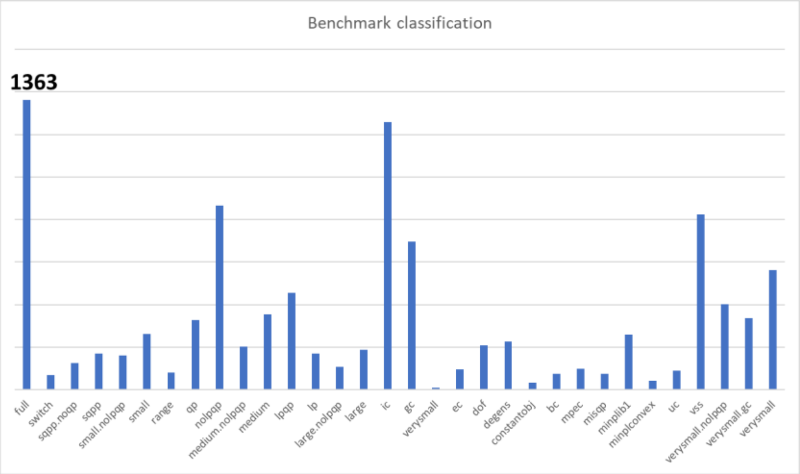
June 29, 2018

BENCHMARK

BENCHMARK IMPROVEMENT

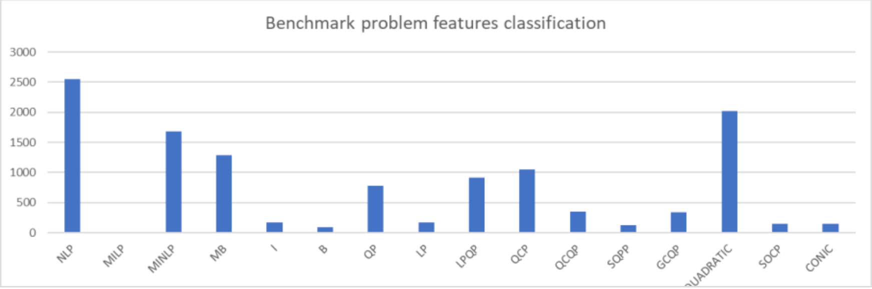
Old benchmark overview :

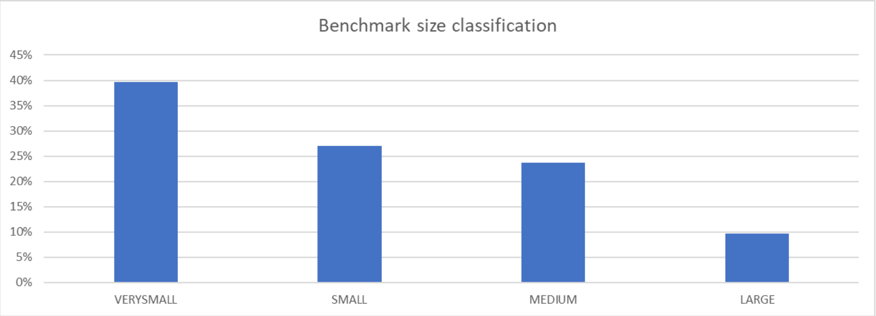
- 1363 instances
- Classified along 40 labels
- Through text lists format : not user-friendly
  - to add tests
  - to add labels, modify them, …



Current status :

- 5000+ instances, 50+ labels
- Categorical labels (small/medium/large) ; problem features (MINLP/QCQP/…)
- Standard benchmarks included
  - Mittelmann
  - GlobalLib-GAMS, MINLPLib2
  - Pglib-opf (for OPF)
- Database format (Excel..)





TEST INTEGRATION

New tests integration process :

- Daily continuous integration
  - Non-regression tests
  - Comparison to a reference run
  - In terms of status, obj, cpu, #iters
- Daily tests report
  - Number of regressions
  - Classified regression table
  - Trends in the improvement of cpu, obj
  - Performance profiles in terms of cpu-time / number of iterations
- Deployment and run on the cluster
  - Use all available resources
  - 1363 instances benchmark is ran in 1h