Artelys Knitro 11.0

SOCP, API, Preconditioner and much more!

September 12, 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

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 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 (~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

NLP ALGORITHMS FEATURES

Features	Interior-Point/Direct	Interior-Point/Conjugate-Gradient	Sequential Linear Quadratic Programming	Sequential Quadratic Programming
Large scale	++ (sparse)	++ (sparse or dense)	+	
Expensive evaluations			+	++
Warm-start	+	+	++	++
Least square problems	++	++	+	+
Globalization technique	Line-search/Trust-region	Trust-region	Trust-region	Line-search/Trust-region
Linear solver	Lapack QR or MA27 or MA57 or MKL PARDISO or **MA86** or **MA97**			
LP solver	-	-	Clp (incl.) or Xpress/Cplex (not incl.)	
QP solver	-	-	-	IP/Direct or IP/CG or SLQP

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

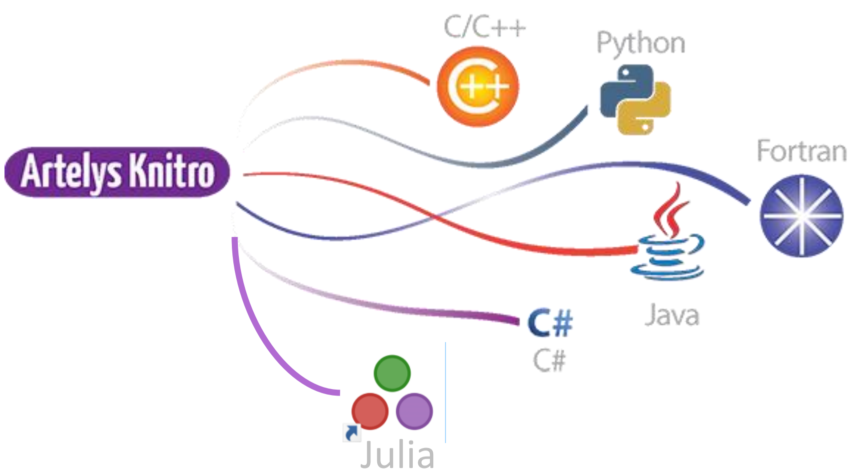
- Economic consulting firms
- Financial institutions
- Mechanical engineering companies
- Oil & Gas companies
- Regulator & Policy maker
- Software developers
  - ↳ Used as a third-party optimization engine

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

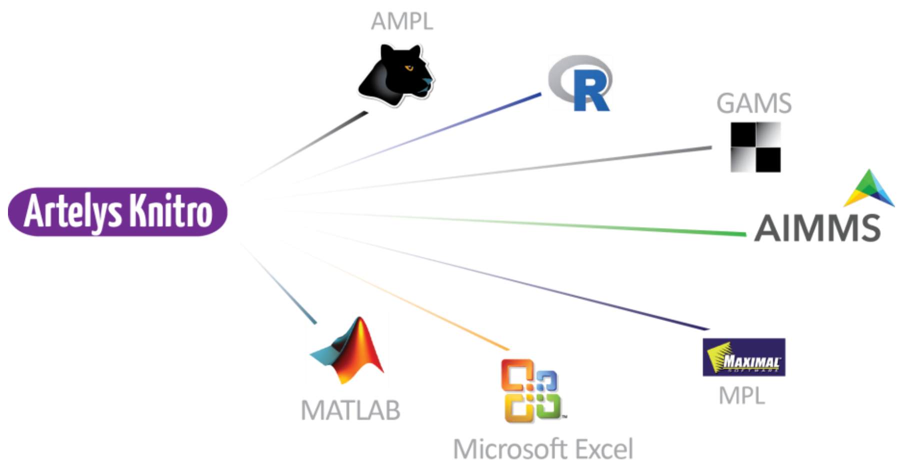
ARTELYS KNITRO: OVERVIEW

Interfaces

- Programming languages



- Modeling languages



Supported platforms

Windows

Windows 32-bit, 64-bit

Linux 64-bit

macOS 64-bit

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
- Preconditioning for all classes of problems
  - **Preconditioning** can now be used for problems with **equality** and inequality constraints

3

- 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

SECOND ORDER CONE CONSTRAINTS

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

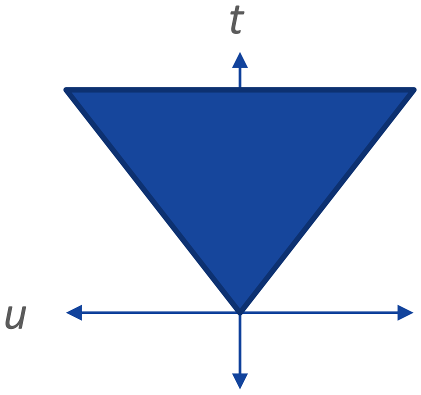
$$\left\{ \begin{bmatrix} t \\ u \end{bmatrix} \middle| u \in R^{k-1}, t \in R^1, \|u\| \leq t \right\}$$

- 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 moves beyond SOCP (more general)

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

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

- Handle any nonlinear problem with second order cone constraints, including non-convex
- Extension of existing Knitro Interior/Direct algorithm via new option

bar_conic_enable=1

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

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

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

$$\Downarrow$$

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

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

$$\boxed{\begin{aligned} & g(x) \leq 0, \lambda \geq 0 \\ & H(x) \in K, w \in K \end{aligned}}$$

$$\boxed{\begin{aligned} & g(x).\lambda = 0 \\ & H(x) \circ w = 0 \end{aligned}}$$

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:

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

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

$$\min_x 0.5x_1 + x_2$$
$$\text{s.t.} \quad x_1^2 \leq x_2^2, \; x_2 \geq 0$$

    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

NLP FORM VS. CONIC FORM

- Steiner tree problem *Steiner_model_100.nl*

```
 The problem is identified as a QCQP.
...
Knitro identified 700 second order cone constraints (after the presolve).
```

- Knitro 10.3 (default)

```
 Iter      Objective      FeasError   OptError    ||Step||    CGits
--------  --------------  ----------  ----------  ----------  -------
       0    4.624889e+000   5.940e+000
      10    4.971916e+000   2.303e-002   1.327e-001   2.667e+000        0

         9970   4.679613e+000   5.991e-008   5.960e-002   5.020e-006        11
    9980    4.679613e+000   5.991e-008   5.962e-002   4.837e-006       11
    9990    4.679614e+000   5.991e-008   5.983e-002   4.691e-006        6
   10000    4.679614e+000   5.991e-008   5.996e-002   4.397e-006        6

EXIT: Iteration limit reached. Current point is feasible.

Final Statistics
----------------
Final objective value               =   4.67961355976477e+000
Final feasibility error (abs / rel) =    5.99e-008 / 1.01e-008
Final optimality error  (abs / rel) =    6.00e-002 / 6.00e-002
# of iterations                     =      10000
# of CG iterations                  =      94362
# of function evaluations           =          0
# of gradient evaluations           =          0
# of Hessian evaluations            =          0
Total program time (secs)           =    1156.271 (   1117.266 CPU time)
Time spent in evaluations (secs)    =       0.000

================================================================================
```

NLP FORM VS. CONIC FORM

- Steiner tree problem *Steiner_model_100.nl*

```
 The problem is identified as a QCQP.
...
Knitro identified 700 second order cone constraints (after the presolve).
```

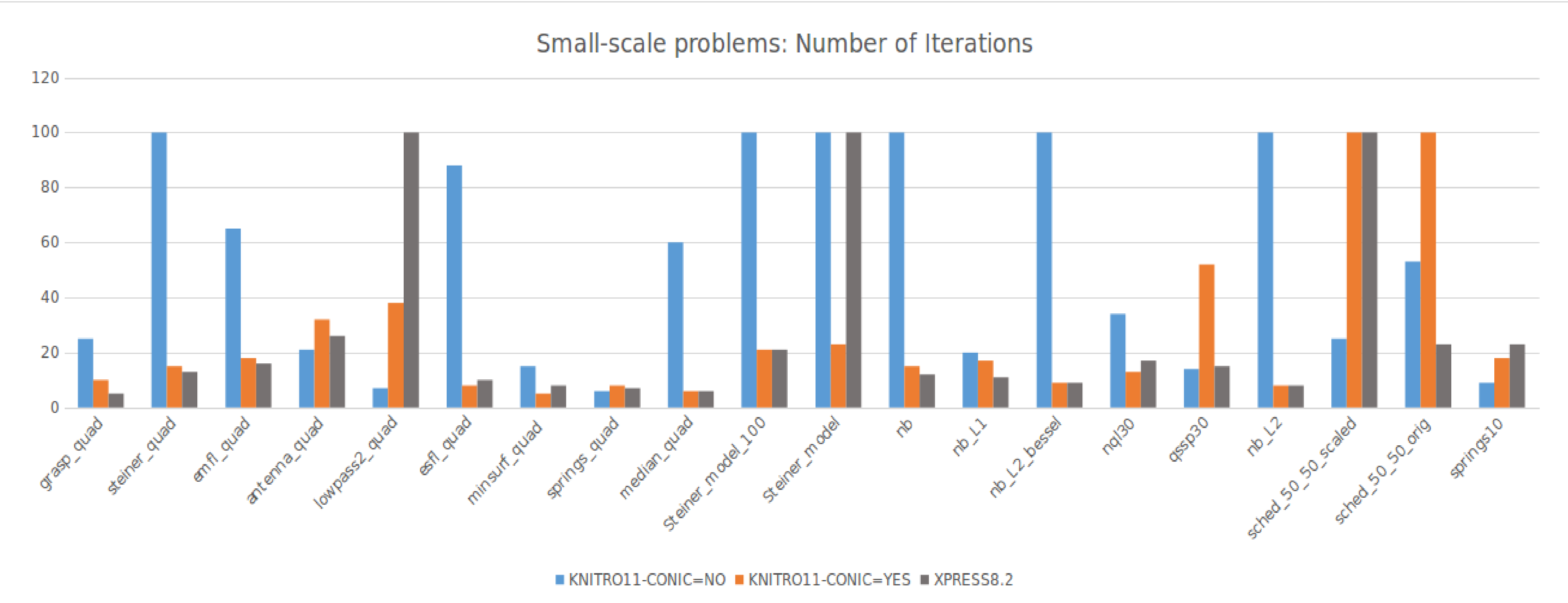
- Knitro 11.0

bar_conic_enable

```
 Iter      Objective      FeasError   OptError    ||Step||    CGits
--------  --------------  ----------  ----------  ----------  -------
       0    4.624889e+000  5.940e+000
      10    4.771003e+000  6.862e-009  2.415e-004  1.494e+000        0
      20    4.678018e+000  4.542e-011  1.146e-006  4.050e-002        0
      21    4.677631e+000  2.029e-011  4.862e-007  2.253e-002        0

EXIT: Locally optimal solution found.

Final Statistics
----------------
Final objective value               =   4.67763066693751e+000
Final feasibility error (abs / rel) =    2.03e-011 / 3.42e-012
Final optimality error  (abs / rel) =    4.86e-007 / 4.86e-007
# of iterations                     =         21
# of CG iterations                  =          0
# of function evaluations           =          0
# of gradient evaluations           =          0
# of Hessian evaluations            =          0
Total program time (secs)           =       1.196 (     1.188 CPU time)
Time spent in evaluations (secs)    =       0.000

===============================================================================
```

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

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

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

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

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

$$y \text{ integer}$$

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

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

s.t. $x_0 x_1 \geq 1$

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

C EXAMPLE

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

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

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

```
// 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

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

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

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

```c
// 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);
```

NEW KNITRO 11.0 API – PERFORMANCE

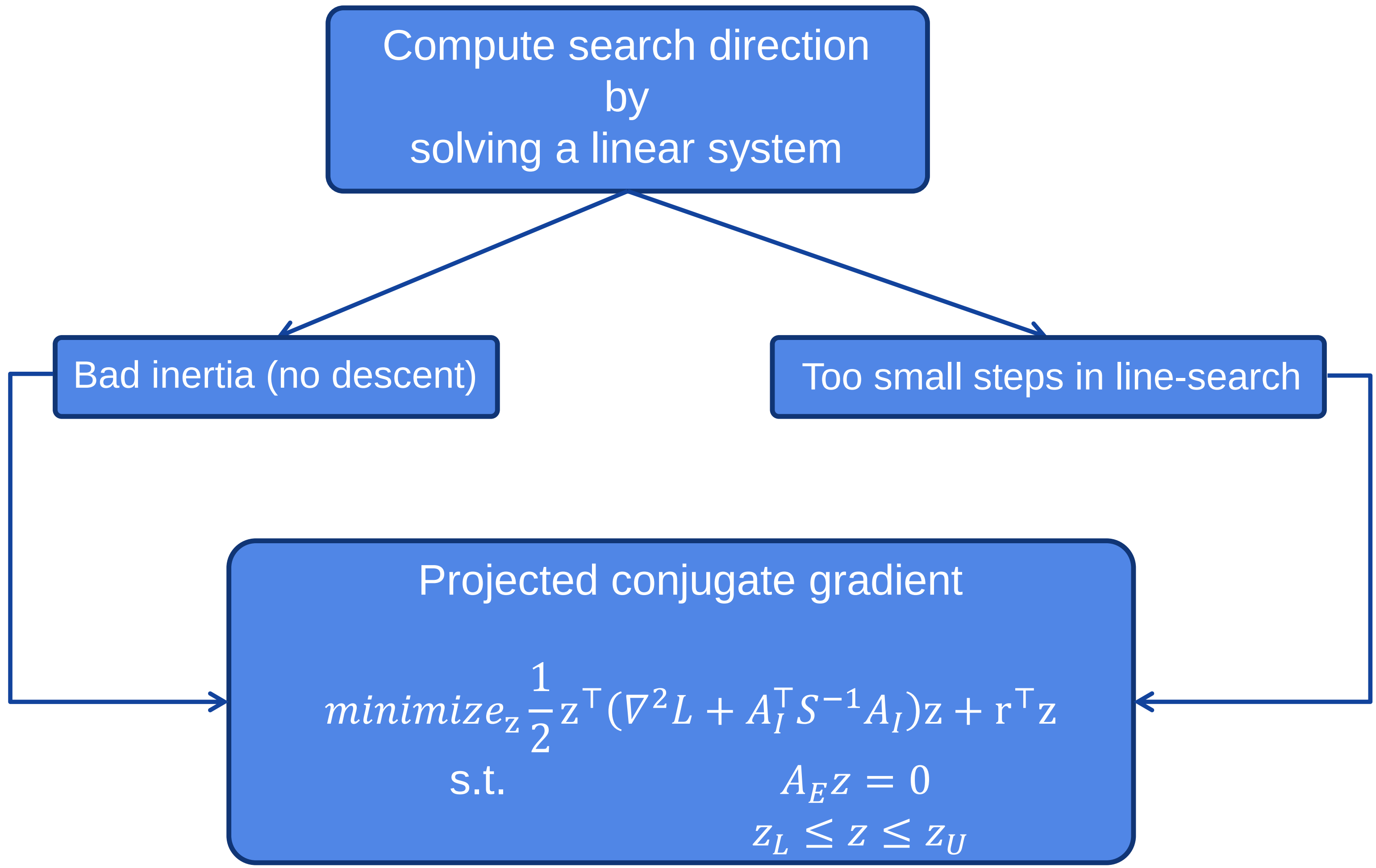
- Comparing old API and new API on some large QCQP models

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



PRECONDITIONING IN KNITRO

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

Compute search direction by solving a linear system

Bad inertia (no descent)

Too small steps in line-search

Projected conjugate gradient

$$
\begin{aligned}
&minimize_z \ \frac{1}{2} z^\top (\nabla^2 L + A_I^\top S^{-1} A_I) z + r^\top z \\
&\text{s.t.} \qquad A_E z = 0 \\
&\qquad\qquad z_L \le z \le z_U
\end{aligned}
$$

Bottleneck: very sensitive to ill-conditionning

PRECONDITIONING IN KNITRO

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

Compute search direction
by
solving a linear system

Bad inertia (no descent)

Too small steps in line-search

Projected conjugate gradient

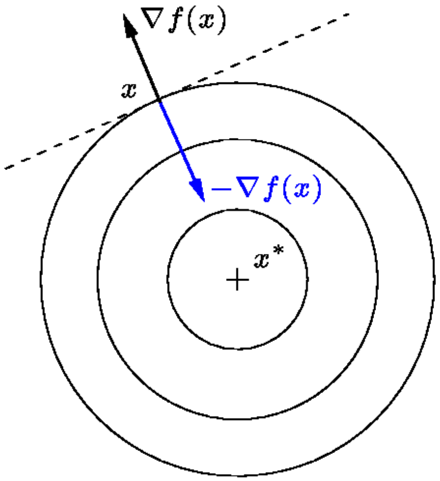
$$minimize_{z} \frac{1}{2} z^{\top}(\nabla^2 L + A_I^{\top} S^{-1} A_I) z + r^{\top} z$$

s.t. $A_E z = 0$

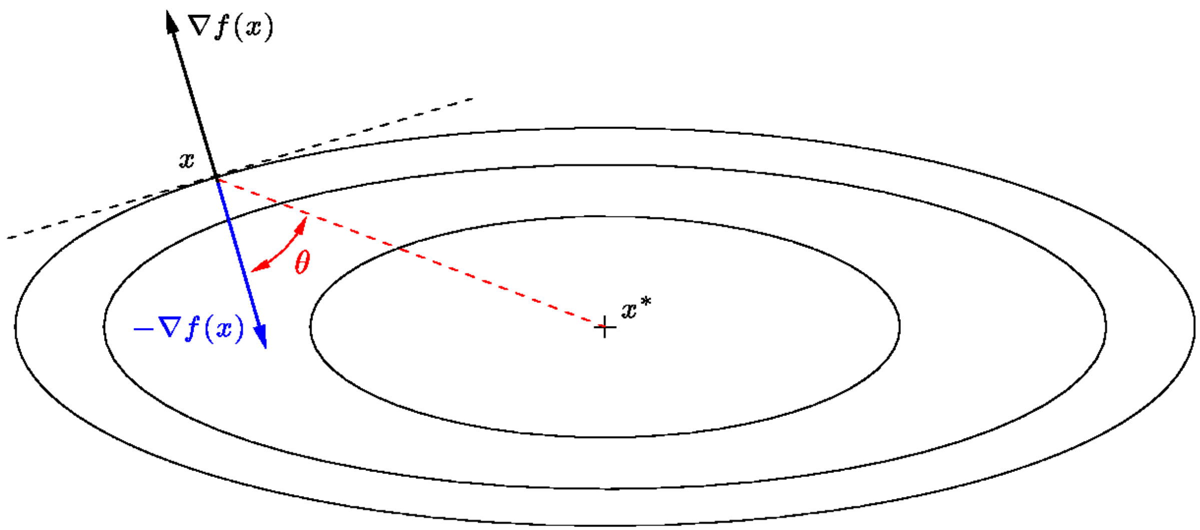
$z_L \leq z \leq z_U$

So just use a preconditioner !

CONDITIONING

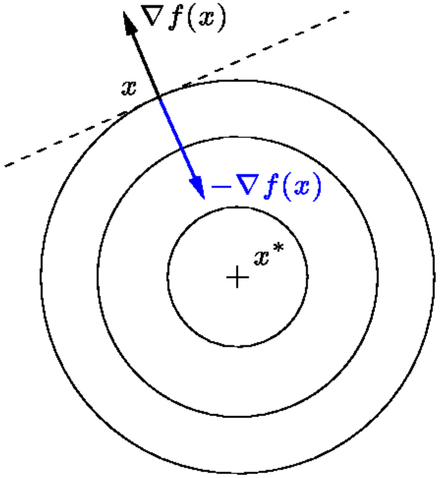


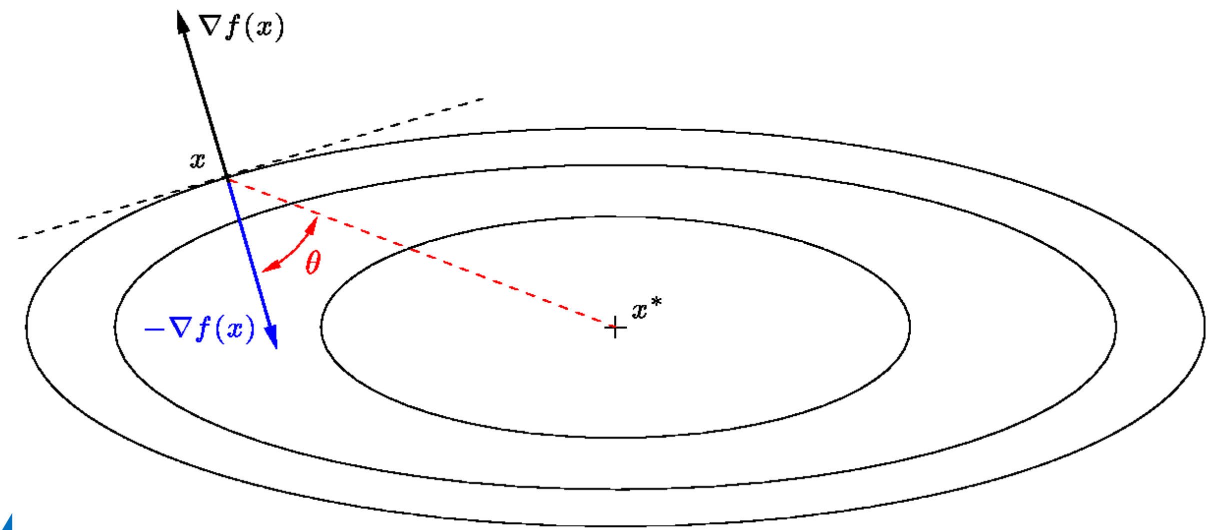
Good conditioning



Bad conditioning

CONDITIONING



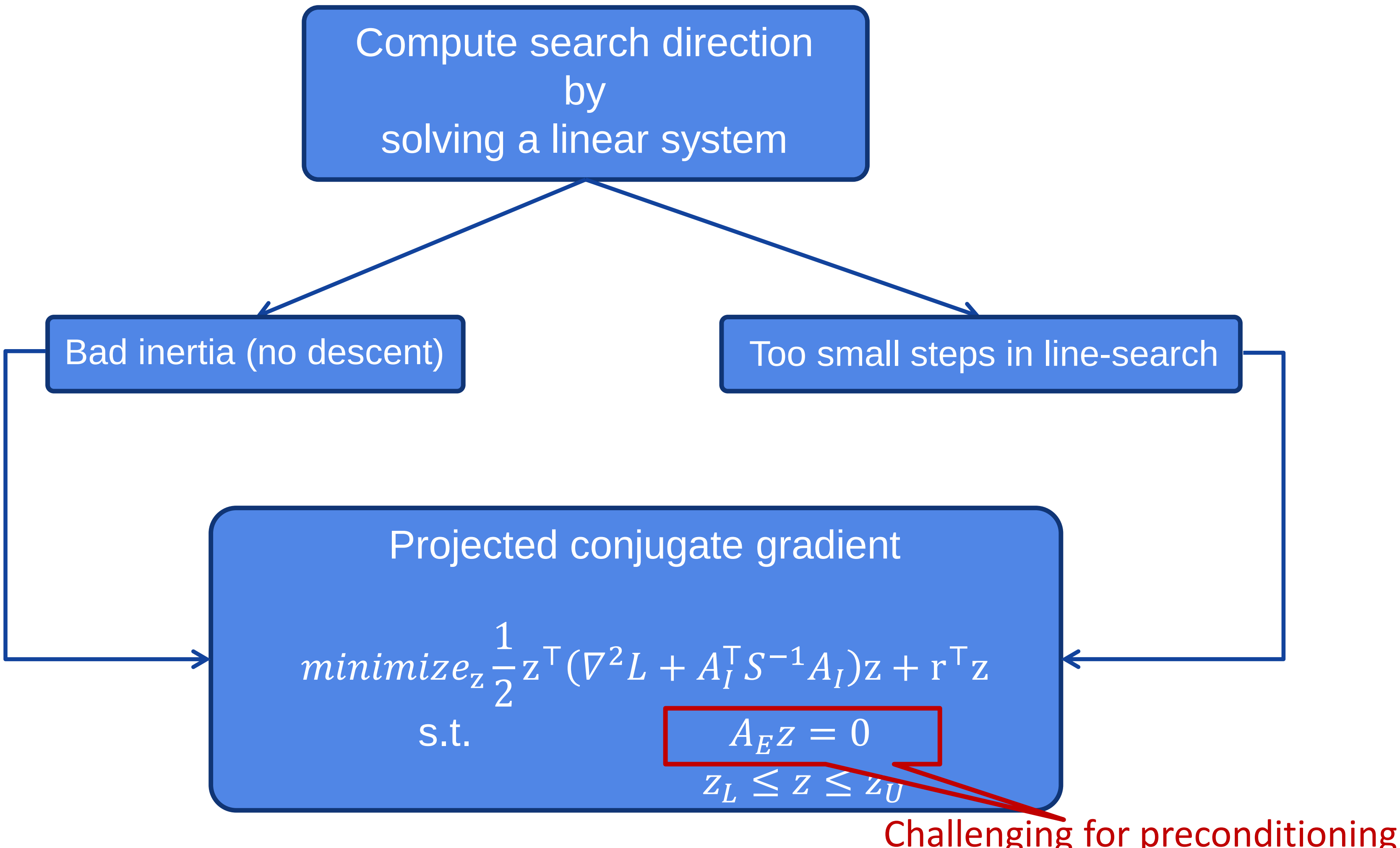


Good conditioning ← Preconditioner ← Bad conditioning

y = Mx

PRECONDITIONING IN KNITRO

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



PRECONDITIONER'S MECHANICS

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

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

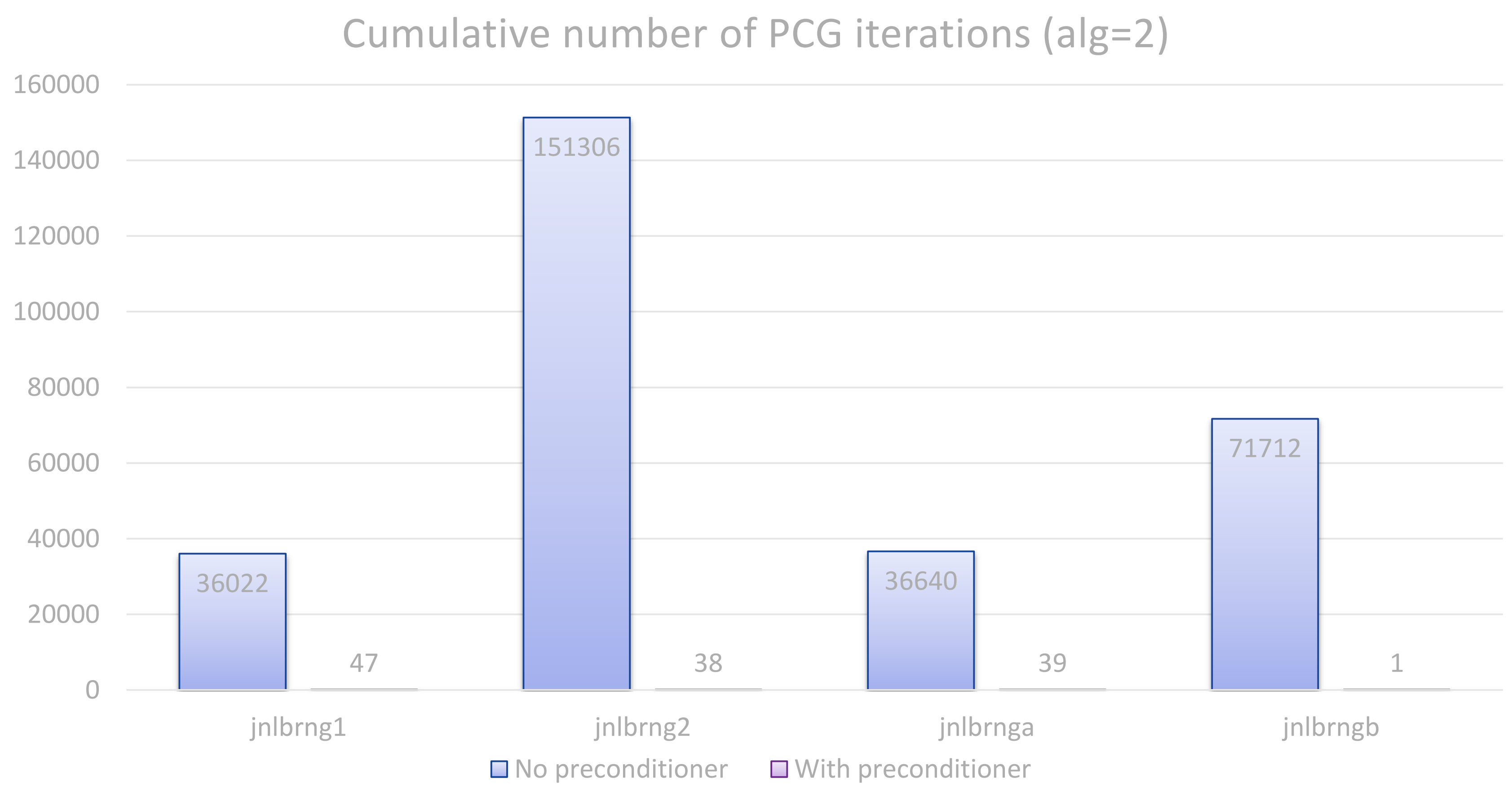
Steps to transform PCG direction so that $\boldsymbol{A_E z = 0}$

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)

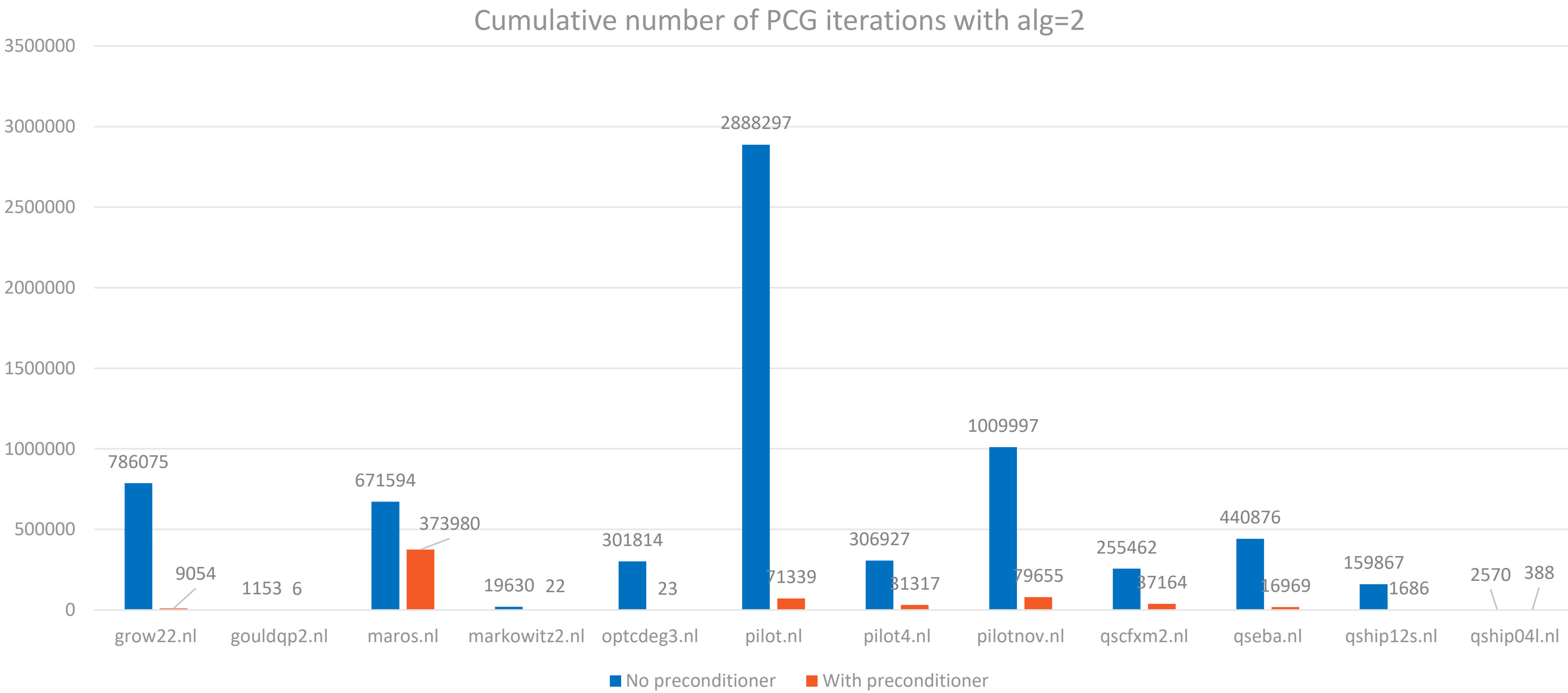
PERFORMANCE IMPROVEMENTS

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



PERFORMANCE IMPROVEMENTS

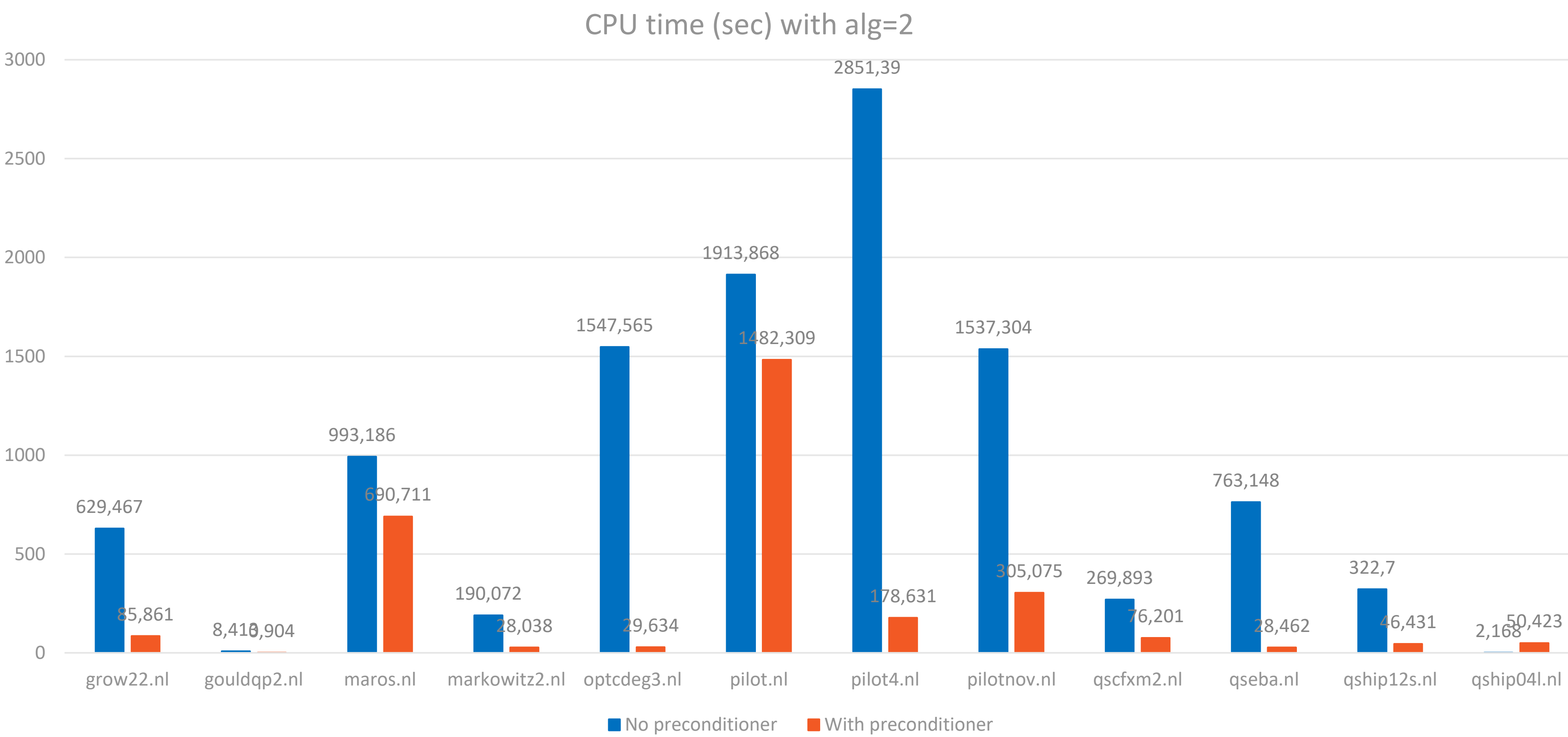
- Nonlinear programs with equality constraints (alg=2, Knitro PCG)



# 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

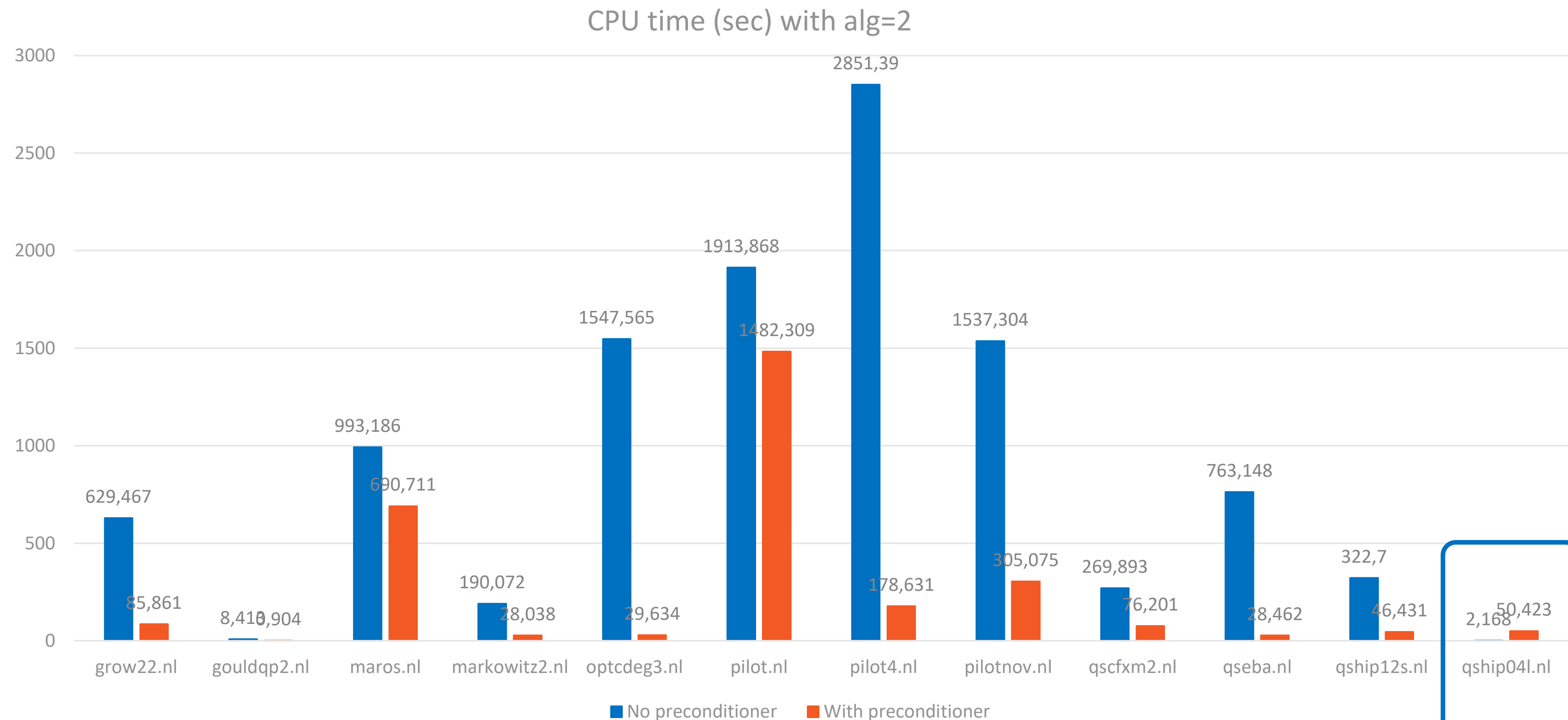
Nonlinear programs with equality constraints (alg=2, Knitro PCG)



# 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

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)

- 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…

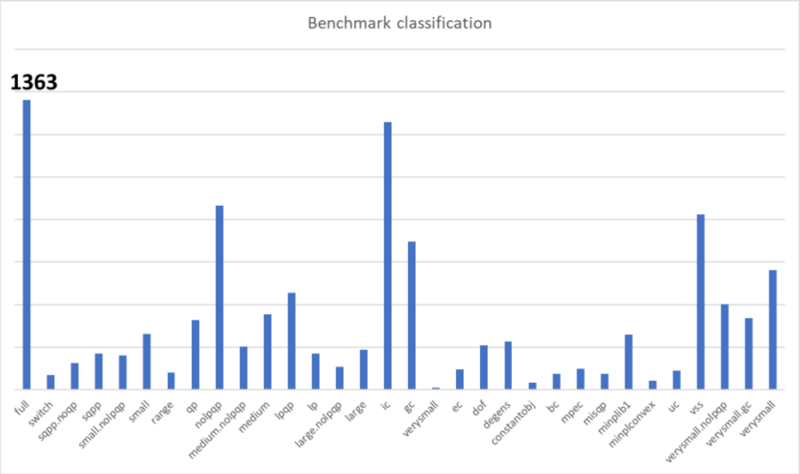
September 12, 2018

BENCHMARK

BENCHMARK IMPROVEMENT

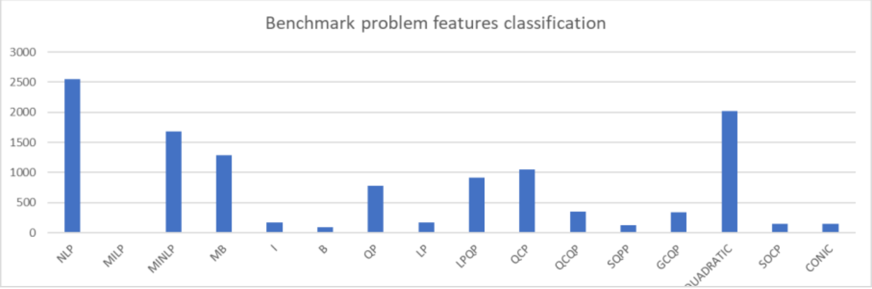
Old benchmark overview :

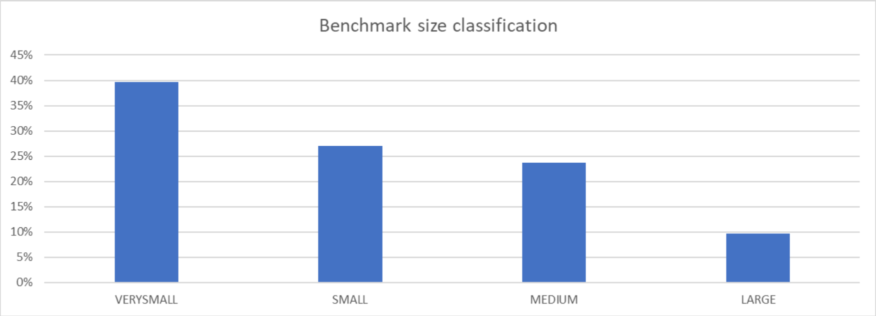
- 1363 instances
- Classified along 40 labels
- Through text lists format : not user-friendly
  - to add tests
  - to add labels, modify them, …



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