QPLIB
a Library of Quadratic Programming Instances

Emiliano Traversi

LIPN, University of Paris 13

ISMP 2015
Pittsburgh

Context

In mathematical optimization, a Quadratic Program (QP) is an optimization problem in which either the objective function or some of the constraints are quadratic functions.

$$
\begin{aligned}
\text{min} \quad & x^T Q_0 x + x^T L_0 + C_0 & \\
\text{s.t.} \quad & x^T Q_i x + x^T L_i + C_i \leq 0 & i = 1, \ldots, n_q \\
& x^T L_i + C_i \leq 0 & i = n_q + 1, \ldots, n \\
& x_j \in \mathbb{R} & j = 1, \ldots, m_r \\
& x_j \in \mathbb{B} & j = m_r + 1, \ldots, m_r + m_b \\
& x_j \in \mathbb{I} & j = m_r + m_b + 1, \ldots, m.
\end{aligned}
$$

Where $Q_0, \ldots, Q_n$ are symmetric matrices of size n, $L_0, \ldots, L_n$ are vectors of size n and $C_0, \ldots, C_n$ are constants.

Context

Instances can be classified according to different:

- objective functions
- constraints
- variables involved.

(Purely) Binary Quadratic Problems

$$
\begin{array}{lllr}
\text{min} & x^T Q_0 x + x^T L_0 + C_0 & & \\
\text{s.t.} & & x^T L_i + C_i \leq 0 & i = 1, \ldots, n \\
 & & x_j \in \mathbb{B} & j = 1, \ldots, m.
\end{array}
$$

Context

Instances can be classified according to different:

- objective functions
- constraints
- variables involved.

Unconstrained Continuous Quadratic Problems

$$
\begin{array}{lll}
\text{min} & x^T Q_0 x + x^T L_0 + C_0 & \\
\text{s.t.} & x_j \in \mathbb{R} & j = 1, \ldots, m.
\end{array}
$$

Context

Instances can be classified according to different:

- objective functions
- constraints
- variables involved.

Continuous Quadratically Constrained Quadratic Problems

$$
\begin{array}{lrr}
\text{min} & x^T Q_0 x + x^T L_0 + C_0 & \\
\text{s.t.} & x^T Q_i x + x^T L_i + C_i \leq 0 & i = 1, \ldots, n_q \\
 & x^T L_i + C_i \leq 0 & i = n_q + 1, \ldots, n \\
 & x_j \in \mathbb{R} & j = 1, \ldots, m.
\end{array}
$$

How to solve a Quadratic Problem?

- Linearization, Outer Approximation
- SDP-based approaches
- Problem-specific approaches
- First/Second Order Methods
- Generic Solvers

At the moment, several generic solvers are available for solving at least one of the mentioned classes of QP. Among the software available, we mention the followings:

- Global Solvers: Antigone, BARON, Couenne, LINDO, SCIP.
- Quadratic Solvers: GloMIQO.
- Binary Quadratic Solvers: FICO Xpress, BiqCrunch, GUROBI, IBM Cplex.
- Max-Cut Solvers: BiqMac.
- Convex Solvers: MOSEK, KNITRO.

- Quadratic Programming Problems have received an increasing amount of attention in recent years, both from theoretical and practical points of view.
- This category of problems models many real-world classes of problems.
- The fact of extending the linear programming with quadratic constraints and objective functions allows a significant increase in the problem modeling power.

- No benchmark is available at the moment for quadratic programming.
- Establishing a benchmark is important, because:
  - it gives more visibility to the topic.
  - it helps to improve the discussion among people working on the subject.
- The QPLIB, aims at being used as reference for the community and the practitioners involved in QP.

(QPLIB Committee)

Alper Atamturk (UC Berkeley), Pietro Belotti (Xpress, FICO), Pierre Bonami (Univ. of Marseille), Samuel Burer (Univ. of Iowa), Sourour Elloumi (ENSIIE), Antonio Frangioni (Univ. of Pisa), Fabio Furini (Univ. of Paris Dauphine), Ambros Gleixner (ZIB Berlin), Nick Gould (Univ. of Oxford), Martin Kidd (Univ. of Bologna), Leo Liberti (LIX), Andrea Lodi (Univ. of Bologna), Ruth Misener (Imperial College London), Nick Sahinidis (Carnegie Mellon Univ.), Frederic Roupin (Univ. of Paris 13), Emiliano Traversi (Univ. of Paris 13), Angelika Wiegele (Univ. of Klagenfurt).

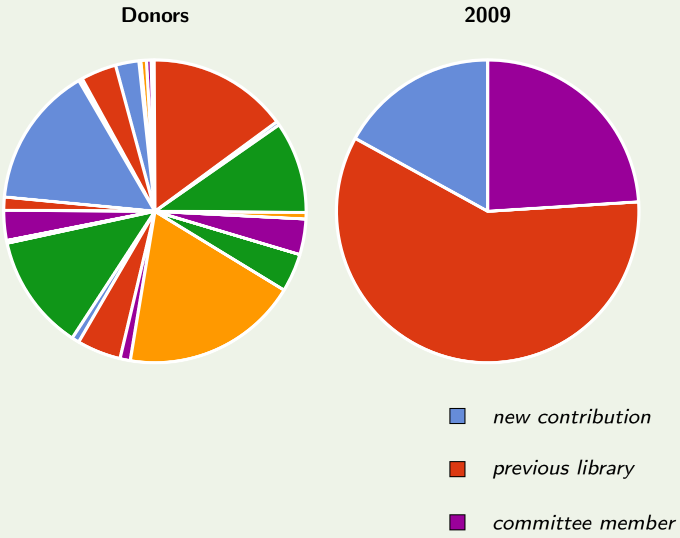
STEP 0: Instance collection

A library needs to be as broad as possible.
Sources of the instances:

- A call for instance has been open for six months.
- Instances coming from existing libraries.
- Instances provided by members of the committee.

At the end of the call:

- *8164 instances.*
- *27 different sources/donors.*
- *4993 instances coming from previous libraries.*
- *1391 instances coming from new contributions.*
- *2044 instances coming from member of the committee*


Donors
2009
new contribution
previous library
committee member

Instance conversion

Several formats used to save the instances: .lp , .gms, .dat .mps and several specific formats (e.g., for the min cut problem).

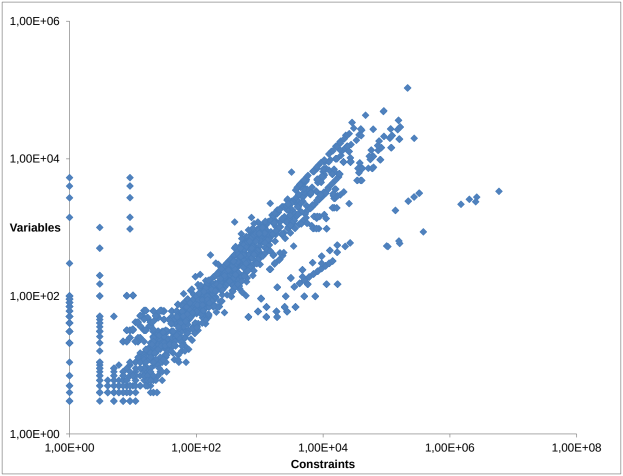
- *We decided to use the GAMS platform to perform all the computational tests.*
- *The General Algebraic Modeling System (GAMS) is a modeling system for mathematical programming and optimization.*
- *It consists of a language compiler and a set of generic solvers.*
- *All the instances are translated in .gms format using either the GAMS platform or one ad-hoc code.*

Step 1: Static Analysis

The following information are collected:

- Dimension, i.e. number of variables and number of constraints.
- Objective function:
  - Convexity of the problem: percentage of negative eigenvalues
  - quadratic, linear or none
- Constraints:
  - Number of quadratic constraints.
  - Number of linear constraints.
  - Number of nonzero entry in linear constraints.
  - Number of nonzero entry in quadratic constraints.
- Variables:
  - Number of continuous variables.
  - Number of integer variables.
  - Number of binary variables.

Instance Size



Convexity of the objective function

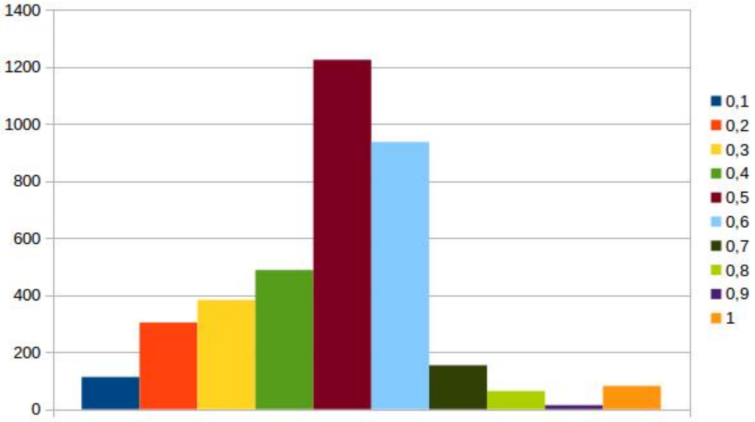


Figure: Percentage of negative eigenvalues

Subdivision into categories

It is important to classify one instances according to its basic features:

Instances are classified according to the following characteristics:

- *objective function: nonexistent, linear, convex quadratic, nonconvex quadratic.*
- *variables: continuous, binary, general integer.*
- *constraints: nonexistent, linear, convex quadratic, nonconvex quadratic.*

Starting set

		No Cont Var		Cont Var			
Obj. Funct.	Constr	Bin Var	Int Var	Bin Var	empty	Int Var	Tot
Lin	Lin	0	0	0	30	0	30
	Quad, Conv	0	60	22	81	112	275
	Quad, NonConv	114	28	2286	222	1	2651
Quad, Conv	Lin	60	28	865	58	8	1019
	Quad, Conv	0	2	5	26	0	33
	Quad, NonConv	0	18	180	31	0	229
Quad NonConv	empty	343	0	0	1	0	344
	Lin	1791	0	16	416	1	2224
	Quad, Conv	0	0	0	424	2	426
	Quad, NonConv	61	0	4	863	5	933
	Tot	2369	136	3378	2152	129	8164

Purely Binary Non-Convex	1791
Purely Binary Convex Quad Constr	0
Purely Binary Non-Convex Quad Constr	175
Purely Continuous Convex	195
Purely Continuous Non-Convex	1957
Mixed integer Convex	901
Mixed integer Non-Convex	17
Mixed Integer Convex Quad Constr	201
Mixed Integer Non-Convex Quad Constr	2524
TOT	8164

Computational Analysis

- The static analysis is not enough to identify the hardness of the instances.
- An empirical way for testing the hardness of one instance is the time needed to solve it.
- We decided to use a broad set of solvers to test the computational hardness of the instances.

Computational Analysis

- *A first round of test is performed on all the instances.*
- *At the moment there is not a solver considered as the state of the art for quadratic problems.*
- *We used the following set of solvers: BARON, Couenne, FICO Xpress, GloMIQO, GUROBI, IBM Cplex, KNITRO, LINDO, MOSEK, SCIP.*
- *A time limit of 30 seconds is imposed.*

- *For each solver and instance, the following information are collected:*
  - *Best primal solution.*
  - *Best dual solution.*
  - *Computing time.*

- *In order to ensure the reproducibility and the fairness of the tests:*
  - *identical nodes has been used for all the tests.*
  - *the same number of cores has been assigned to each job.*
  - *an external time limit is imposed on each job.*

Goal of the preliminary selection is to obtain a smaller, easy to handle, subset of instances.

- *As first step, we excluded from the test bed all the instances solved by at least 30% of the solvers.*
- *As second step, for each donor/source, we excluded all the instances with similar performances and similar characteristics.*

Preliminary Computational Analysis - Filter

Starting set	*8164 Instances*	
	$\Downarrow$	
	6012 Discr. Inst.	*2152 Cont. inst.*
Filter on TL 30 sec	$\Downarrow$	
	2743 Discr. Inst.	$\Downarrow$
Filter on similarity	$\Downarrow$	
	604 Discr. Inst.	*211 Cont. inst.*

Discrete Instances

		Constraints				
		empty	Lin	Quad, Conv	Quad, NonConv	Tot
Obj. Funct.	Lin	0	0	45	311	356
	Quad, Conv	1	49	0	3	52
	Quad, NonConv	29	155	0	12	196
	Tot	29	204	45	326	604

Continuous Instances

		Constraints			
		Lin	Quad, Conv	Quad, NonConv	Tot
Obj. Funct.	Lin	0	19	64	83
	Quad, Conv	12	0	5	17
	Quad, NonConv	24	33	54	111
	Tot	36	52	123	211

Final Computational Analysis

- A final round of tests is performed with a time limit of 10800 seconds.
- A subset of 7 solvers is used for the last round of tests.

Hardness of the instances

Instances are classified according to the number of solver able to solve them within the time limit:

- `Open`: none of the solvers.
- `Hard`: less than 25% of the solvers.
- `Medium`: between 25% and 50% of the solvers.
- `Easy`: more than 75% of the solvers.

Discrete Instances			
`Open`	`Hard`	`Medium`	`Easy`
152	113	291	105

Continuous Instances			
`Open`	`Hard`	`Medium`	`Easy`
55	18	108	31

Final Computational Analysis

- All the `Open` and `Hard` instances are kept.
- The `Medium` and `Easy` instances are filtered.

Final Set - Continuous instances

	Constr			
Obj. Funct	Lin	Quad, Conv	Quad, NonConv	Tot
Lin	0	9	44	53
Quad, Conv	7	0	3	10
Quad, NonConv	18	19	38	75
Tot	25	28	85	138

Final Set - Discrete instances

		No Cont Var		Cont Var		
Obj. Funct.	Constr	Bin Var	Int Var	Bin Var	Int Var	Tot
Lin	Quad, Conv	0	11	4	8	23
	Quad, NonConv	43	5	176	3	227
Quad, Conv	Lin	12	0	16	0	28
	Quad, NonConv	0	0	1	0	1
Quad NonConv	empty	24	0	0	0	24
	Lin	84	0	11	1	96
	Quad, NonConv	8	0	2	1	11
	Tot	171	16	210	13	410

Conclusion

- We provided a library tuned for Quadratic Programming Problems.

Conclusion

- We provided a library tuned for Quadratic Programming Problems.
- A beta version of the library will be available online soon:

  `www.lamsade.dauphine.fr/QPlib2014/`

Conclusion

- We provided a library tuned for Quadratic Programming Problems.
- A beta version of the library will be available online soon:

  `www.lamsade.dauphine.fr/QPlib2014/`
- A free, non proprietary format .qplib will be introduced.

Conclusion

- We provided a library tuned for Quadratic Programming Problems.
- A beta version of the library will be available online soon:

  `www.lamsade.dauphine.fr/QPlib2014/`
- A free, non proprietary format .qplib will be introduced.
- Feedbacks are more than welcome!

THANK YOU!