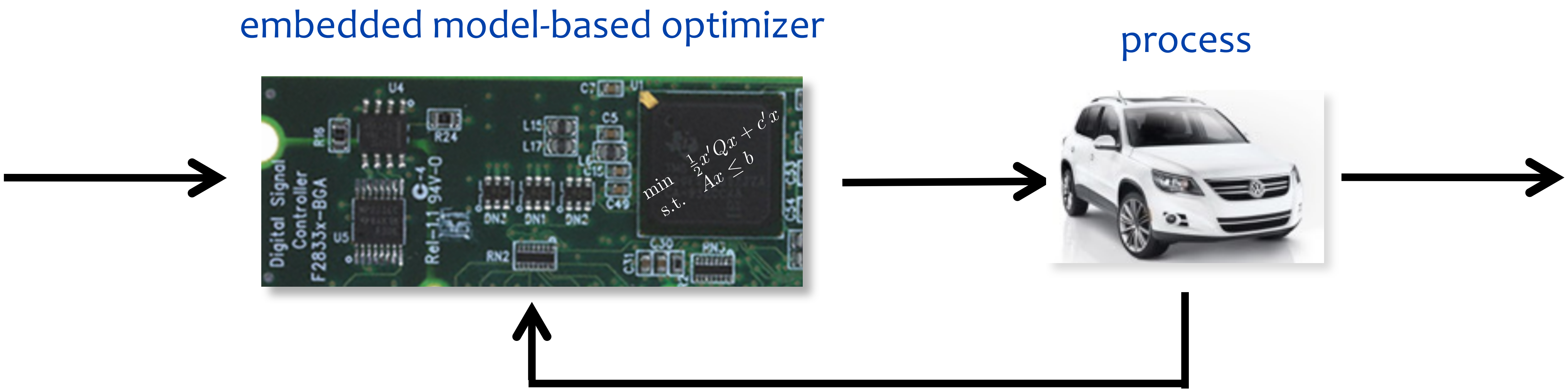
Solving Mixed-Integer Quadratic Programs via Nonnegative Least Squares

Alberto Bemporad

http://cse.lab.imtlucca.it/~bemporad/

IMT INSTITUTE FOR ADVANCED STUDIES LUCCA

MPC in a production environment



Requirements for production:



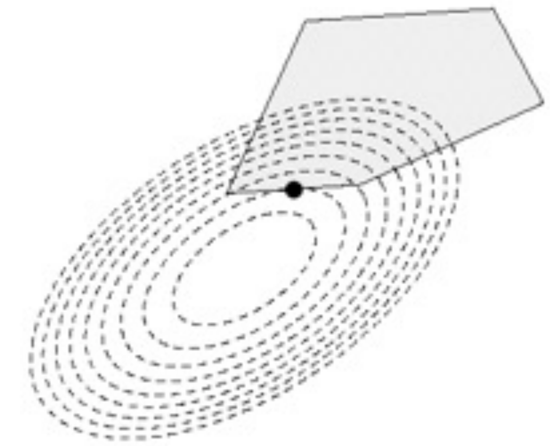


1. **Speed (throughput)**: solve optimization problem within sampling interval
2. **Robust** with respect to finite-precision arithmetics
3. Be able to run on **limited hardware** (e.g., 150 MHz) with **little memory**
4. **Worst-case execution time** must be (tightly) estimated
5. **Code simple** enough to be validated/verified/certified
   (in general, it must be understandable by production engineers)

Embedded Liner MPC

- Linear MPC requires solving a **Quadratic Program (QP)**

$$\begin{array}{ll}\min\limits_{z} & \frac{1}{2}z'Hz + x'(t)F'z + \cancel{\frac{1}{2}x'(t)Yx(t)} \\ \text{s.t.} & Gz \leq W + Sx(t)\end{array} \qquad z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

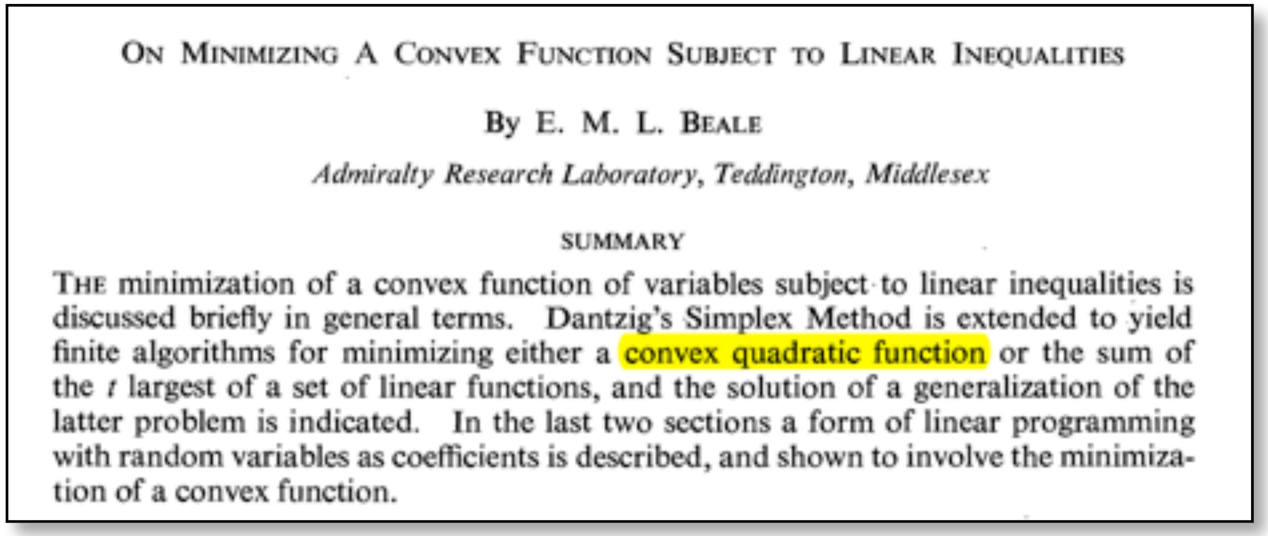


- Several algorithms exist to solve the QP **on-line** given $x(t)$:

  active set (AS), interior point (IP), gradient projection (GP), alternating direction method of multipliers (ADMM), proximal methods, ...

- or **offline** (*explicit MPC*)

A rich set of good QP algorithms is available today



ON MINIMIZING A CONVEX FUNCTION SUBJECT TO LINEAR INEQUALITIES

By E. M. L. BEALE

Admiralty Research Laboratory, Teddington, Middlesex

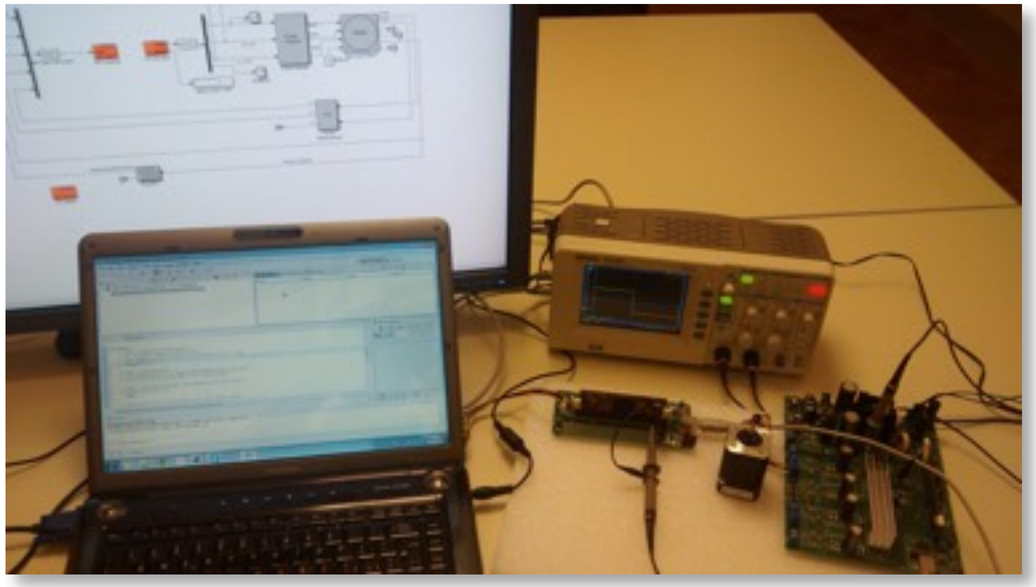
SUMMARY

THE minimization of a convex function of variables subject to linear inequalities is discussed briefly in general terms. Dantzig's Simplex Method is extended to yield finite algorithms for minimizing either a convex quadratic function or the sum of the t largest of a set of linear functions, and the solution of a generalization of the latter problem is indicated. In the last two sections a form of linear programming with random variables as coefficients is described, and shown to involve the minimization of a convex function.

(Beale, 1955)

Experiments with embedded QP

- Experimental setup:
  - PC with MATLAB/Simulink
  - RS232 adapter
  - TMS320F28335 Experimenter Kit



var × constr.	GPAD	AS	ADMM	FBN
4×16	332 μs (18)	120 μs (3)	1.42 ms (62)	208 μs (2)
8×24	1.1 ms (22)	446 μs (5)	4 ms (77)	396 μs (2)
12×32	2.59 ms (27)	1.19 ms (7)	8.25 ms (82)	652 μs (2)

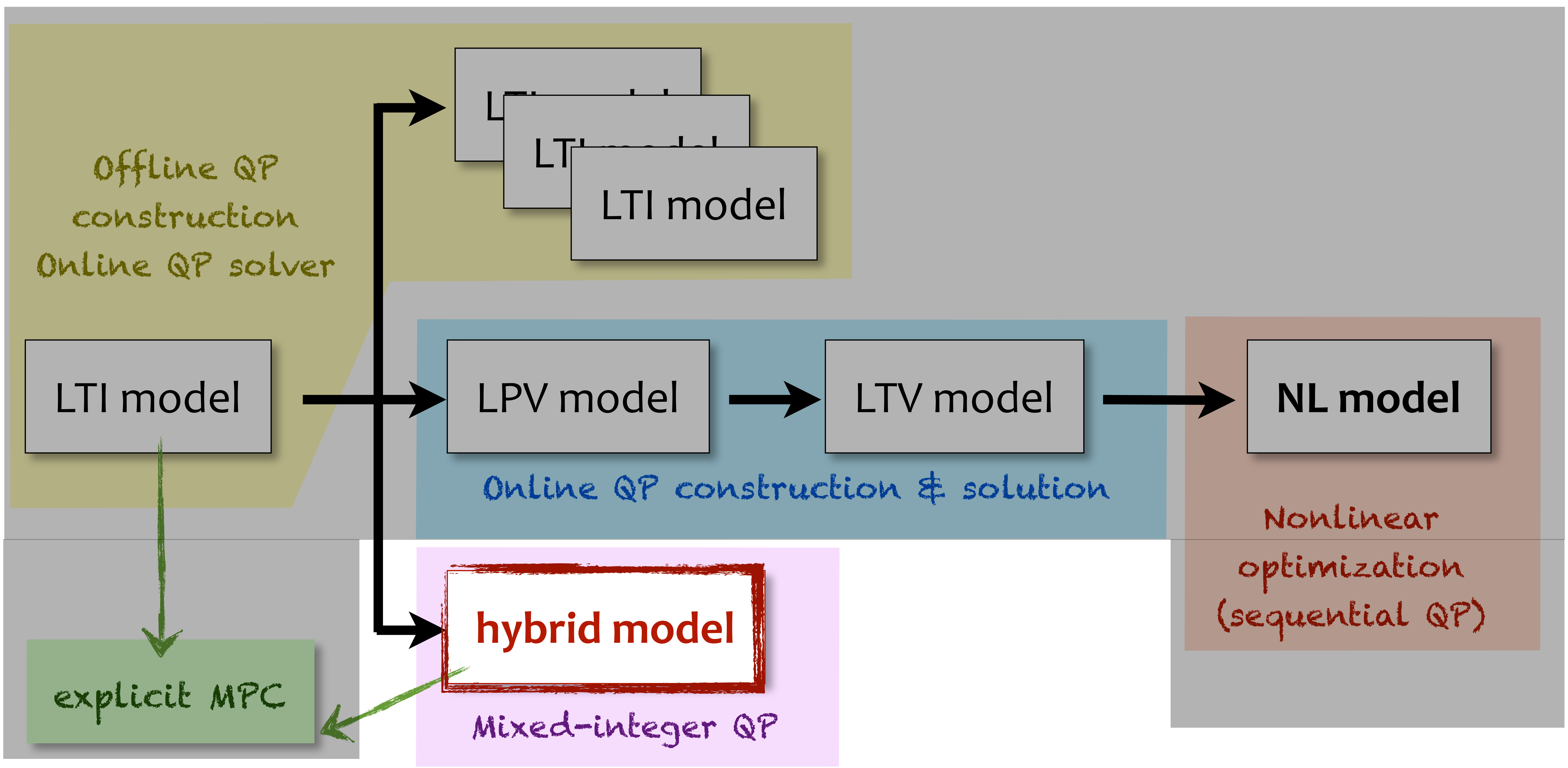
- **Active set** (AS) methods usually are best on small problems:
  - excellent quality solutions within few iterations
  - less sensitive to preconditioning (= behavior is more predictable)
  - no need for advanced linear algebra packages

* GPAD = Dual Accelerated Gradient Projection (Patrinos, Bemporad, 2014)

* FBN = Forward-Backwards Netwon (proximal method) (Patrinos, Guiggiani, Bemporad, 2014)

Beyond embedded linear MPC ...

- One Linear Time-Invariant (LTI) model is often not enough for MPC (nonlinearities, model dependence on external signals, binary decisions)



Embedded hybrid MPC

$$
\begin{array}{ll}
\min_\xi & \displaystyle\sum_{k=0}^{N-1} y_k' Q y_k + u_k' R u_k \\
\text{s.t.} & \left\{\begin{array}{rcl}
x_{k+1} & = & A x_k + B_1 u_k + B_2 \delta_k + B_3 z_k + B_5 \\
y_k & = & C x_k + D_1 u_k + D_2 \delta_k + D_3 z_k + D_5 \\
E_2 \delta_k + E_3 z_k & \leq & E_4 x_k + E_1 u_k + E_5
\end{array}\right.
\end{array}
$$

Mixed Logical Dynamical model

(Bemporad, Morari, 1999)

$$
\begin{array}{ll}
\min_\xi & \frac{1}{2}\xi' H \xi + x'(t) F \xi \\
\text{s.t.} & G\xi \leq W + S x(t)
\end{array}
$$

Mixed Integer Quadratic Program (MIQP)

- Optimization vector $\xi \in \mathbb{R}^{n_c} \times \{0,1\}^{n_b}$ has both **real** and **binary** values

Can we embed a MIQP solver in a control module ?

MIQP for embedded hybrid MPC

- Excellent free and commercial solvers available for MIQP and MILP (Gurobi, CPLEX, GLPK, Xpress-MP, CBC) …

- … but none of them is really tailored to embedded applications

- Contributions to embedded MIQP algorithms:

– Branch and bound (B&B) + dual active set solvers (Leyffer, Fletcher, 1998) (Axehill, Hansson, 2006)

– Branch-and-bound + fast embedded interior point solvers with offline pre-processing heuristics (Frick, Domahidi, Morari, 2015)

This contribution: a new B&B MIQP algorithm exploiting a novel **dual active set method for QP** based on **nonnegative least squares**

Why Least Squares ?

The **Least Squares (LS)** problem is probably the most studied problem in numerical linear algebra


(Legendre, 1805)


(Gauss, <= 1809)

$$v = \arg\min \|Av - b\|_2^2$$

In MATLAB: >> `v=A\b % (1 character)`

- **Nonnegative Least Squares (NNLS):**

$$\begin{array}{ll} \min_v & \|Av - b\|_2^2 \\ \text{s.t.} & v \geq 0 \end{array}$$

- **Partially Nonnegative Least Squares (PNNLS):**

$$\begin{array}{ll} \min_{v,u} & \|Av + Bu - c\|_2^2 \\ \text{s.t.} & v \geq 0, \ \ u \text{ free} \end{array}$$

PNNLS = pseudo-inverse of B
+ solve NNLS

Active-set method for Nonnegative Least Squares

(Lawson, Hanson, 1974)

$$\begin{array}{ll}\min_v & \|Av-b\|_2^2\\ \text{s.t.} & v\geq 0\end{array}$$

1) $\mathcal{P}\leftarrow\emptyset$, $v\leftarrow 0$;
2) $w\leftarrow A'(Av-b)$;
3) **if** $w\geq 0$ **or** $\mathcal{P}=\{1,\ldots,m\}$ **then go to** Step 11;
4) $i\leftarrow\arg\min_{i\in\{1,\ldots,m\}\setminus\mathcal{P}} w_i$, $\mathcal{P}\leftarrow\mathcal{P}\cup\{i\}$;
5) $y_{\mathcal{P}}\leftarrow\arg\min_{z_{\mathcal{P}}}\|((A')_{\mathcal{P}})'z_{\mathcal{P}}-b\|_2^2$, $y_{\{1,\ldots,m\}\setminus\mathcal{P}}\leftarrow 0$;
6) **if** $y_{\mathcal{P}}\geq 0$ **then** $v\leftarrow y$ **and go to** Step 2;
7) $j\leftarrow\arg\min_{h\in\mathcal{P}:\ y_h\leq 0}\left\{\frac{v_h}{v_h-y_h}\right\}$;
8) $v\leftarrow v+\frac{v_j}{v_j-y_j}(y-v)$;
9) $\mathcal{I}\leftarrow\{h\in\mathcal{P}: v_h=0\}$, $\mathcal{P}\leftarrow\mathcal{P}\setminus\mathcal{I}$;
10) **go to** Step 5;
11) $v^*\leftarrow v$; **end**.

Algorithm: *While maintaining primal var v feasible, keep switching active set until dual var w is also feasible*

- NNLS algorithm is very simple (`750 chars in Embedded MATLAB`), the key operation is to solve a standard LS problem at each iteration (via QR, LDL', or Cholesky factorization)

Solving QP's via nonnegative least squares

(Bemporad, IEEE TAC, 2016)

- Use NNLS to solve strictly convex QP

$$\min_z \ \tfrac{1}{2}z'Qz + c'z \quad \text{s.t.} \ Gz \le g$$

QP

complete the squares

$$u \triangleq Lz + L^{-T}c$$

$$Q = L'L$$
$$M = GL^{-1}$$
$$d = b + GQ^{-1}c$$

$$\min_u \ \tfrac{1}{2}\|u\|^2 \quad \text{s.t.} \ Mu \le d$$

Least Distance Problem

(Lawson, Hanson, 1974)

$$\min_y \ \tfrac{1}{2}\left\| \begin{bmatrix} M' \\ d' \end{bmatrix} y + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\|_2^2 \quad \text{s.t.} \ y \ge 0$$

Nonnegative Least Squares

residual $= 0$?

yes: QP problem infeasible

no:

$$z^* = -\frac{1}{1+d'y^*}L^{-1}M'y^* - Q^{-1}c$$

retrieve primal solution

Relation between NNLS problem and dual QP

- Relation with dual QP problem:

$$\min_z \ \tfrac{1}{2}z'Qz + c'z \quad \text{s.t.} \ Gz \leq g$$

$\xrightarrow{\text{LDP}}$

$$\min_y \ \frac{1}{2}\left\|\begin{bmatrix} M' \\ d' \end{bmatrix} y + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right\|_2^2 \quad \text{s.t.} \ y \geq 0$$

duality

$MM' = GL^{-1}L^{-T}G' = GQ^{-1}G'$

$$\min_\lambda \ \tfrac{1}{2}\lambda'(MM')\lambda + d'\lambda \quad \text{s.t.} \ \lambda \geq 0$$

can be unbounded if primal QP is infeasible

$$\min_y \ \tfrac{1}{2}y'(MM' + dd')y + d'y \quad \text{s.t.} \ y \geq 0$$

always bounded by -½
(cost = -½ if primal QP is infeasible)

- If primal QP is feasible, we can prove that

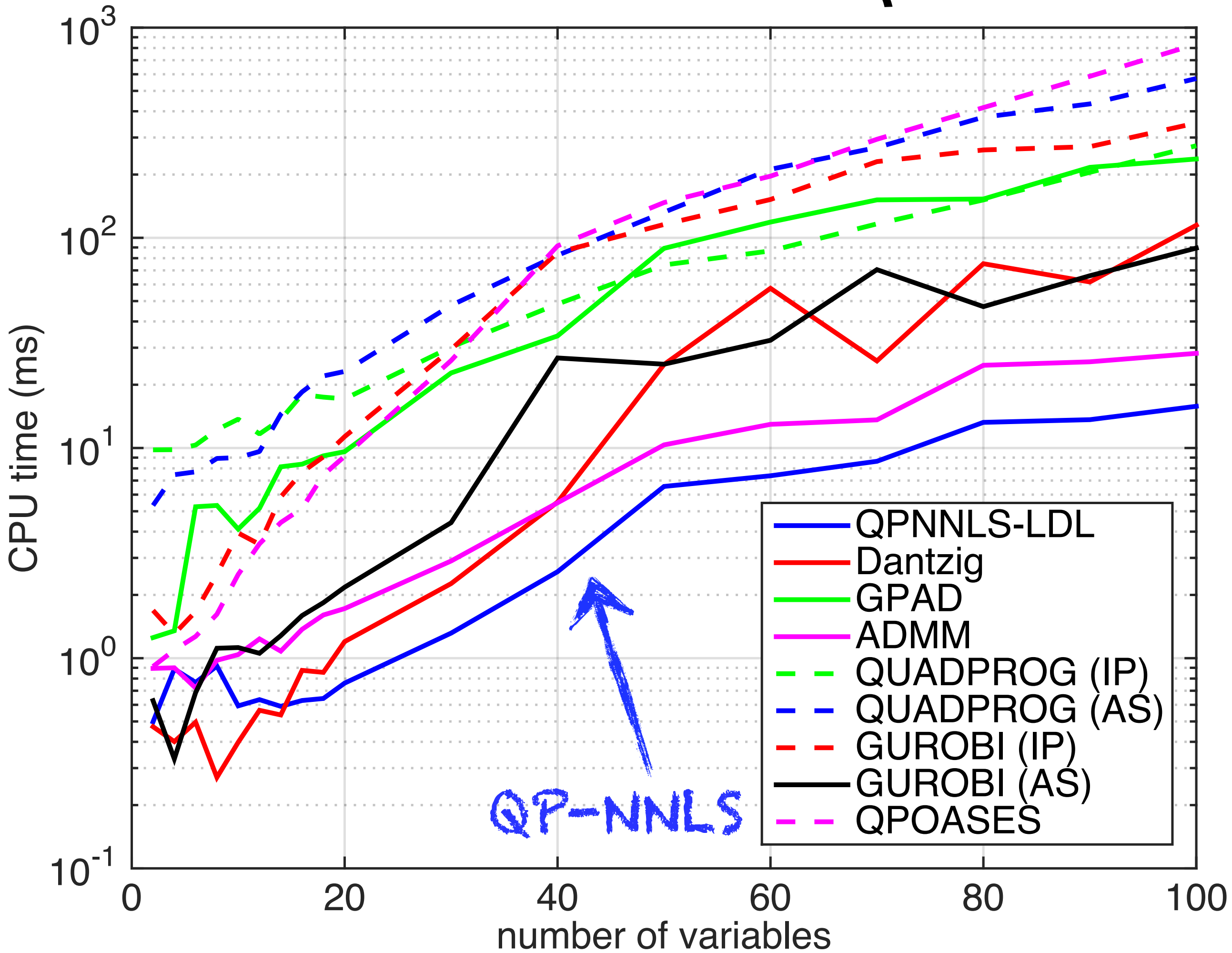
$$\lambda^* = \frac{1}{1 + d'y^*}y^*$$

- During NNLS iterations, dual QP cost can be easily obtained from y

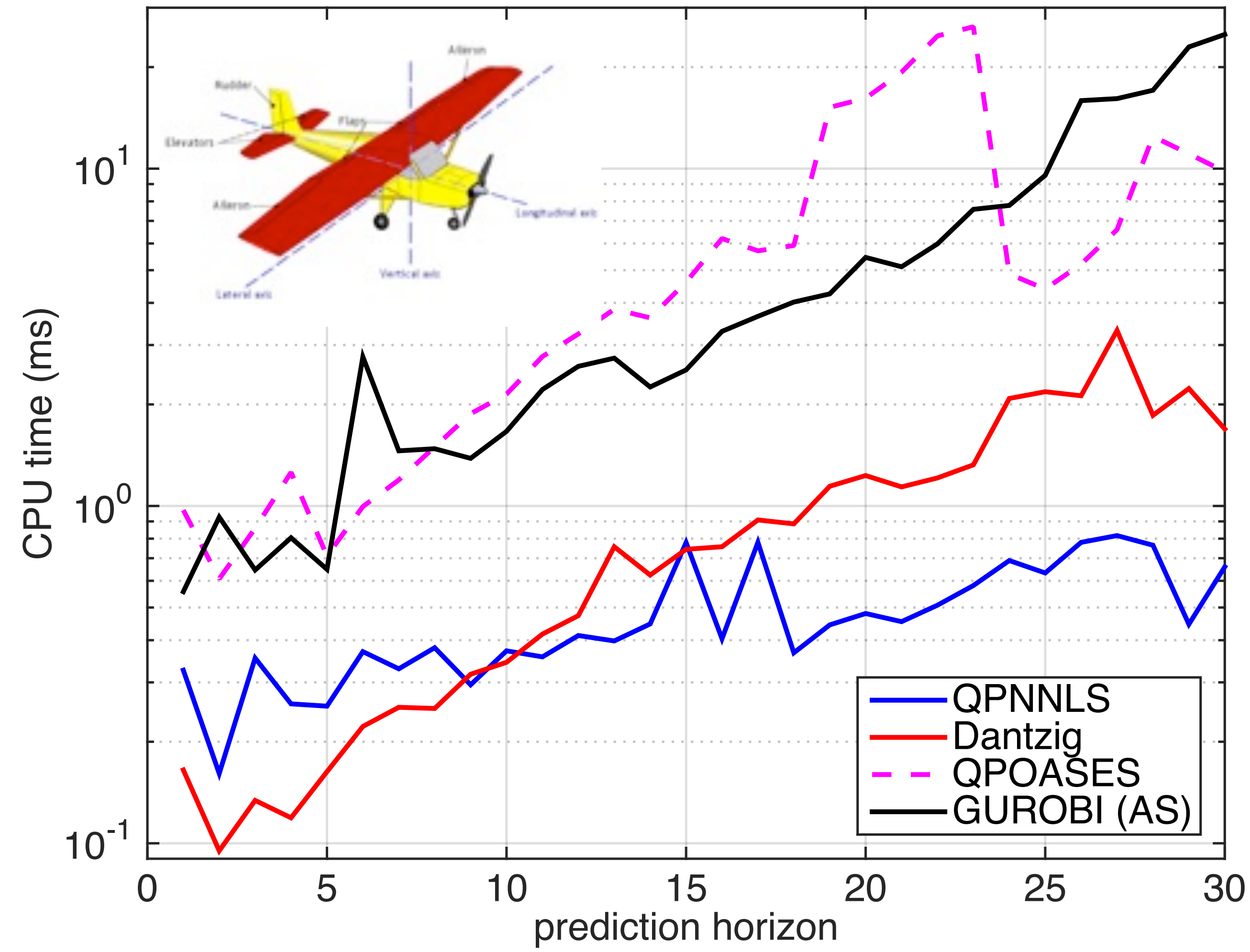
Solving QP via NNLS: Numerical results

(Bemporad, IEEE TAC, 2016)

worst-case over 100 random QP instances



worst-case occurred during entire simulation*



* Step t=0 not considered for QPOASES not to penalize the benefits of the method with warm starting

- A rather **fast** and relatively **simple-to-code** QP solver !

NNLS for multiparametric QP

- A variety of mpQP solvers is available

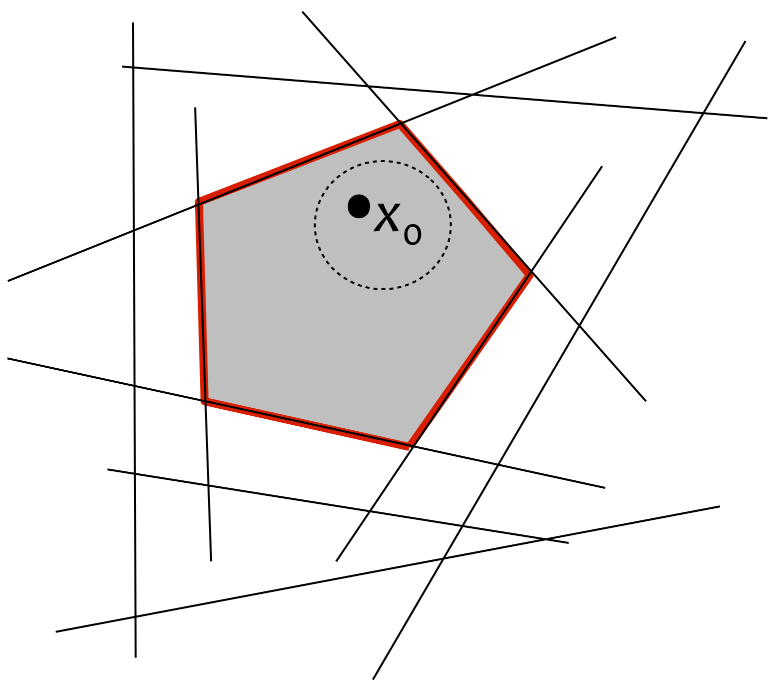
(Bemporad *et al.*, 2002) (Baotic, 2002)
(Tøndel, Johansen, Bemporad, 2003)
(Spjøtvold *et al.*, 2006)(Patrinos, Sarimveis, 2010)

- Most computations are spent in **operations on polyhedra** (=critical regions)

$$\begin{array}{rcl} \widehat{G}z^*(x) & \leq & \widehat{W} + \widehat{S}x \\ \widetilde{\lambda}^*(x) & \geq & 0 \end{array}$$

feasibility of primal solution
feasibility of dual solution

- checking **emptiness of polyhedra**
- removal of **redundant inequalities**
- checking **full-dimensionality** of polyhedra



- All such operations are usually done via **linear programming (LP)**

NNLS for multiparametric QP

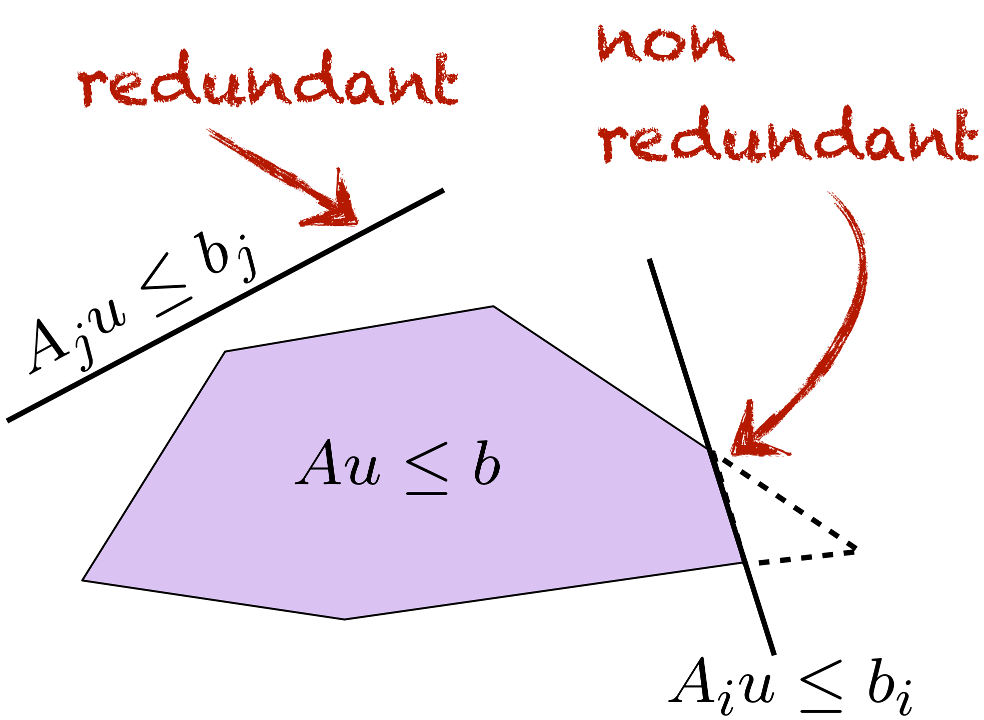
(Bemporad, IEEE TAC 2015)

- Key result:

A polyhedron $P = \{u \in \mathbb{R}^n : \; Au \leq b\}$
is nonempty iff the PNNLS problem

$$(v^*, u^*) = \arg\min_{v,u} \; \|v + Au - b\|_2^2 \quad \text{s.t.} \quad v \geq 0, u \text{ free}$$

has zero residual $\|v^* + Au^* - b\|_2^2 = 0$



- Numerical results on elimination of redundant inequalities:

m	NNLS	LP
2	0.0006	0.0046
4	0.0019	0.0103
6	0.0038	0.0193
8	0.0071	0.0340
10	0.0111	0.0554
12	0.0178	0.0955
14	0.0263	0.1426
16	0.0357	0.1959

random polyhedra of $\mathbb{R}^m$ with $10m$ inequalities

NNLS = compiled Embedded MATLAB
LP = compiled C code (GLPK)

CPU time = seconds (this Mac)

- Many other polyhedral operations can be also tackled by NNLS

NNLS for solving mpQP problems

- New mpQP algorithm based on **NNLS** + **dual QP formulation** to compute active sets and deal with degeneracy (Bemporad, IEEE TAC, 2015)

- Comparison with:
  - Hybrid Toolbox (Bemporad, 2003)
  - Multiparametric Toolbox 2.6 (with default opts) (Kvasnica, Grieder, Baotic, 2006)

q	m	Hybrid Tbx	MPT	NNLS
4	2	0.0174	0.0256	0.0026
4	3	0.0263	0.0356	0.0038
4	4	0.0432	0.0559	0.0061
4	5	0.0650	0.0850	0.0097
4	6	0.0827	0.1105	0.0126
8	2	0.0347	0.0396	0.0050
8	3	0.0583	0.0680	0.0092
8	4	0.0916	0.0999	0.0140
8	5	0.1869	0.2147	0.0322
8	6	0.3177	0.3611	0.0586
12	2	0.0398	0.0387	0.0054
12	3	0.1121	0.1158	0.0191
12	4	0.2067	0.2001	0.0352
12	5	0.6180	0.6428	0.1151
12	6	1.2453	1.3601	0.2426
20	2	0.1029	0.0763	0.0152
20	3	0.3698	0.2905	0.0588
20	4	0.9069	0.7100	0.1617
20	5	2.2978	1.9761	0.4395
20	6	6.1220	6.2518	1.2853

- Included in MPC Toolbox 5.0 (≥R2014b)

The MathWorks (Bemporad, Morari, Ricker, 1998-2015)

NNLS for solving MIQP problems

- We consider a MIQP problem of the following form

$$\begin{aligned} \min_z \quad & V(z) \triangleq \frac{1}{2} z'Qz + c'z \\ \text{s.t.} \quad & \ell \leq Az \leq u \\ & Gz = g \\ & \bar{A}_i z \in \{\bar{\ell}_i, \bar{u}_i\}, \ \ i = 1, \ldots, q \end{aligned}$$

$$Q = Q' \succ 0$$

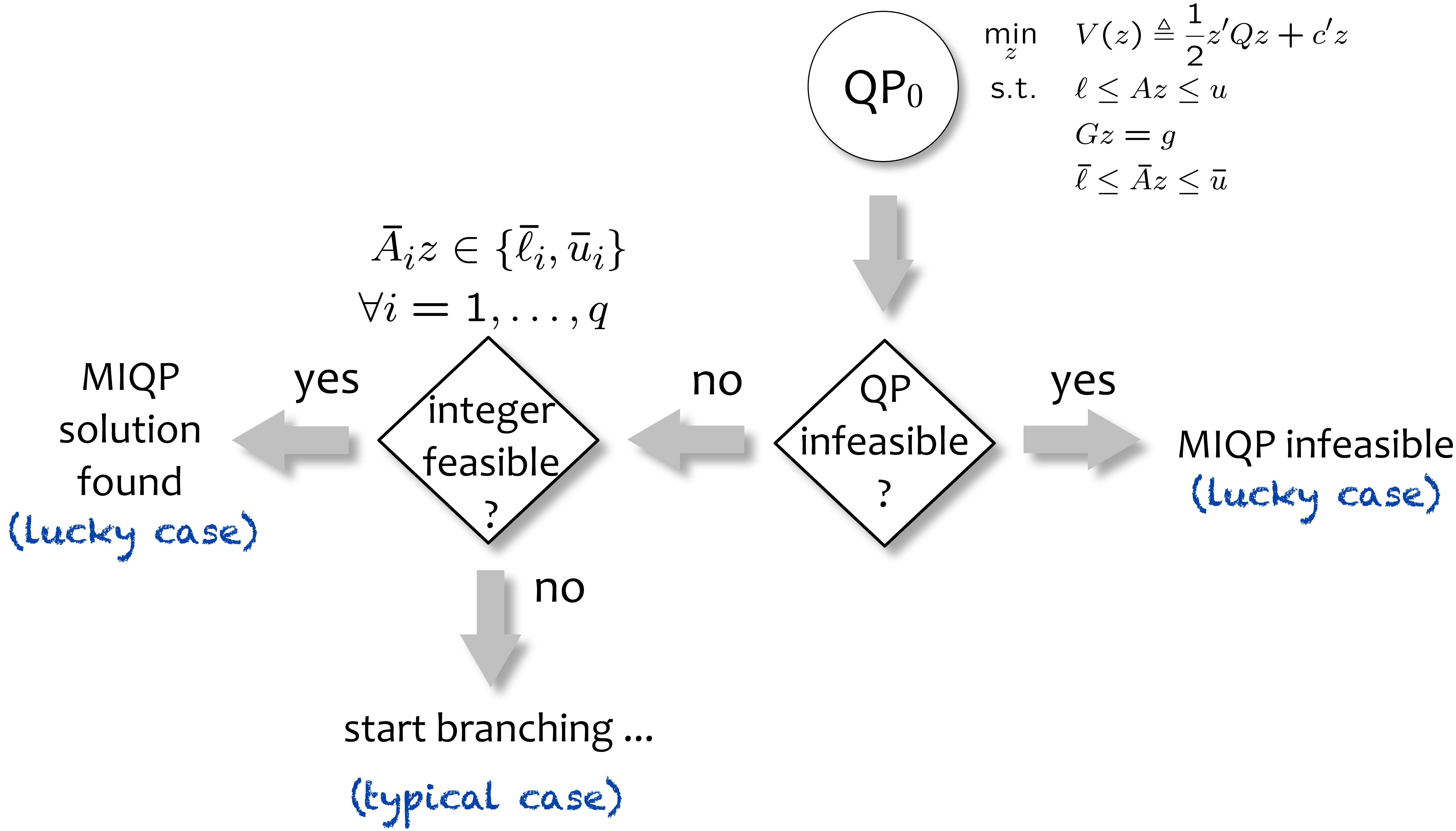
- Binary constraints on z are a special case:

$$\bar{\ell}_i = 0, \ \bar{u}_i = 1, \ \bar{A}_i = [0 \ldots 0 \ 1 \ 0 \ldots 0]$$

- QP algorithm based on NNLS is **extended** here to handle
  - **equality** constraints
  - **bilateral** constraints
  - **warm-starts**

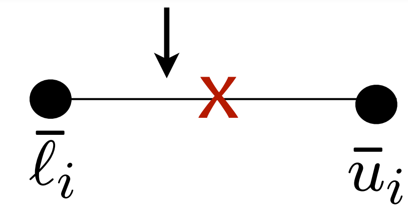
  so to solve MIQP relaxations ($\bar{\ell}_i \leq \bar{A}_i z \leq \bar{u}_i$) very efficiently

- **Branch and bound** scheme:

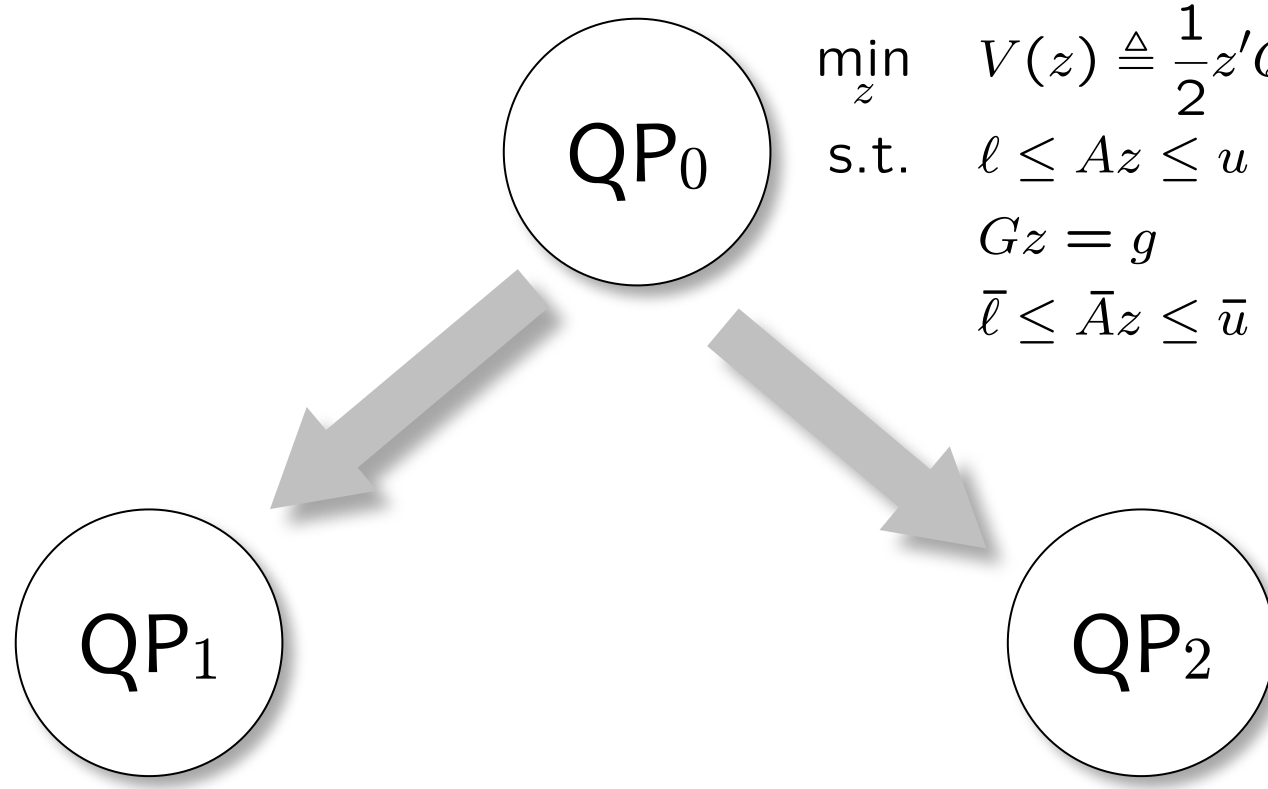


NNLS for solving MIQP problems

- **Branching:** pick up index i such that $\bar{A}_i z$ is closest to $\frac{\bar{\ell}_i+\bar{u}_i}{2}$



- Solve two new QP problems:



QP_0:

$$\begin{array}{ll} \min\limits_{z} & V(z) \triangleq \frac{1}{2}z'Qz + c'z \\ \text{s.t.} & \ell \leq Az \leq u \\ & Gz = g \\ & \bar{\ell} \leq \bar{A}z \leq \bar{u} \end{array}$$

QP_1:

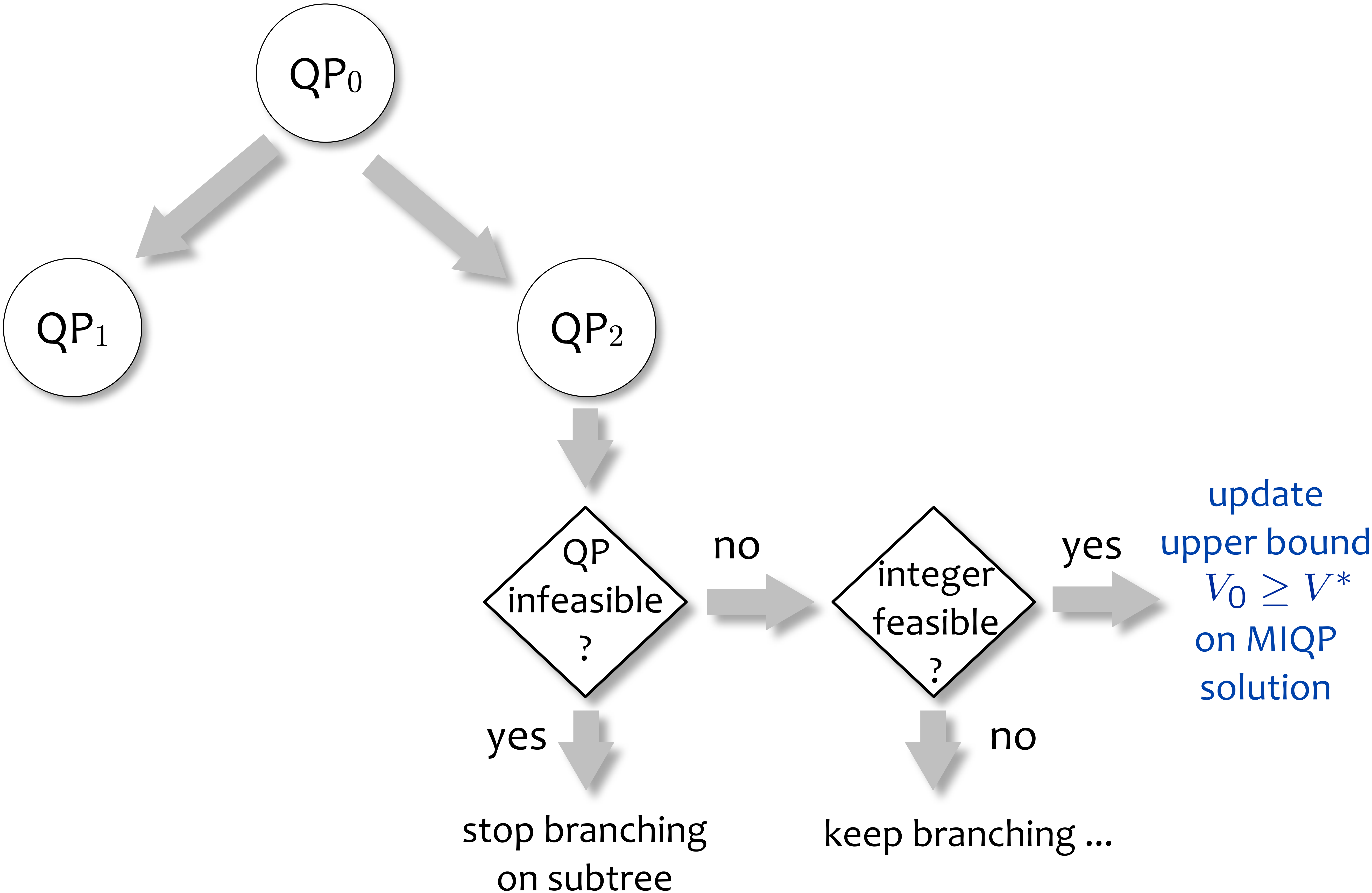
$$\begin{array}{ll} \min\limits_{z} & V(z) \triangleq \frac{1}{2}z'Qz + c'z \\ \text{s.t.} & \ell \leq Az \leq u \\ & Gz = g \\ & A_i z = \bar{\ell}_i \\ & \bar{\ell}_j \leq \bar{A}_j z \leq \bar{u}_j, \quad j \neq i \end{array}$$

QP_2:

$$\begin{array}{ll} \min\limits_{z} & V(z) \triangleq \frac{1}{2}z'Qz + c'z \\ \text{s.t.} & \ell \leq Az \leq u \\ & Gz = g \\ & A_i z = \bar{u}_i \\ & \bar{\ell}_j \leq \bar{A}_j z \leq \bar{u}_j, \quad j \neq i \end{array}$$

Warm start from previous solution of QP_0 heavily exploited in solving QP_1, QP_2 !

NNLS for solving MIQP problems



NNLS for solving MIQP problems

- **Lower bound** V on optimal solution of QP relaxation are immediately available during the iterations of the QP algorithm

→ stop solving QP relaxation if $V \geq V_0$

This saves a lot of
QP active set
iterations !

- When no further branching is possible, either the MIQP problem is recognized infeasible or the optimal solution has been found

Numerical results

- Worst-case CPU time on random **MIQP** problems:

n	m	q	NNLS_{LDL}	NNLS_{QR}	GUROBI	CPLEX
10	5	2	2.3	1.2	1.4	8.0
10	100	2	5.7	3.3	6.1	31.4
50	25	5	4.2	6.1	14.1	30.1
50	200	10	68.8	104.4	114.6	294.1
100	50	2	4.6	10.2	37.2	69.2
100	200	15	137.5	365.7	259.8	547.8
150	100	5	15.6	49.2	157.2	260.1
150	300	20	1174.4	3970.4	1296.1	2123.9

n = # variables, m = # inequality constraints, no equalities, q = # binary constraints

QP algorithm in compiled Embedded MATLAB code, B&B in interpreted MATLAB code.
CPU time measured on this Mac

NNLS_{LDL} = **recursive LDL' factorization** used to solve least-square problems in QP solver
NNLS_{QR} = **recursive QR factorization** used instead (numerically more robust)

Numerical results

- Worst-case CPU time on random **purely binary QP** problems:

n	m	q	NNLS_{LDL}	NNLS_{QR}	GUROBI	CPLEX
2	10	2	5.1	4.0	0.7	8.4
4	20	4	8.9	4.3	4.5	16.7
8	40	8	19.2	18.0	37.1	14.7
12	60	12	59.7	57.8	82.3	47.9
20	100	20	483.5	457.7	566.8	99.6
25	250	25	110.4	93.3	1054.4	169.4
30	150	30	1645.4	1415.8	2156.2	184.5

- Worst-case CPU time on a **hybrid MPC** problem (Bemporad, Morari, 1999)

N = prediction horizon

MIQP regularized to make Q strictly >0 (--> solution difference is negligible)

$n = 40,\ m = 160,\ q = 10$

N	NNLS_{LDL}	NNLS_{QR}	GUROBI	CPLEX
2	2.2	2.3	1.2	3.0
3	3.4	3.9	2.0	6.5
4	5.0	6.5	2.6	8.1
5	7.6	9.8	3.7	9.0
6	12.3	17.7	4.3	11.0
7	20.5	30.5	5.8	13.1
8	28.9	47.1	7.3	17.3
9	38.8	62.5	9.5	18.9
10	55.4	98.2	10.9	22.4

Conclusions

- Need to **dig into the numerical details of MIQP** to make hybrid MPC suitable for fast embedded applications (when explicit is not possible)

- **Ongoing work:**

  - combine **hybrid modeling** and B&B solver (Bemporad, Giorgetti, 2006)

  - recursive hybrid systems identification for **adaptive** hybrid MPC (Bemporad, Breschi, Piga, submitted)

Is linear MPC really a mature technology for production in fast embedded applications ?

YES !

Is hybrid MPC mature too ?

WE'RE VERY CLOSE …

ODYS Advanced Controls & Optimization