Quadratic optimization

Robert Luce

October 2022

GUROBI OPTIMIZATION

Basic terminology and formulations

Examples of quadratic functions

- Univariate function:

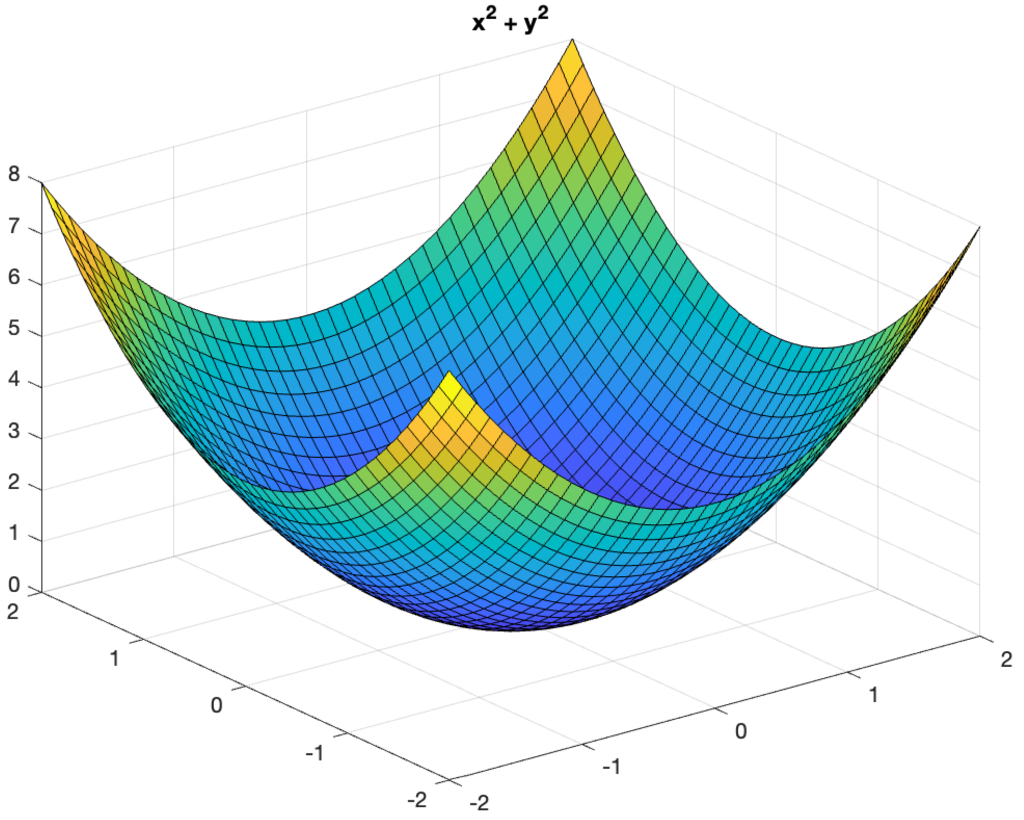
$$f\colon \mathbb{R} \longrightarrow \mathbb{R}, \qquad f(x) = x^2 + 2x + 1$$

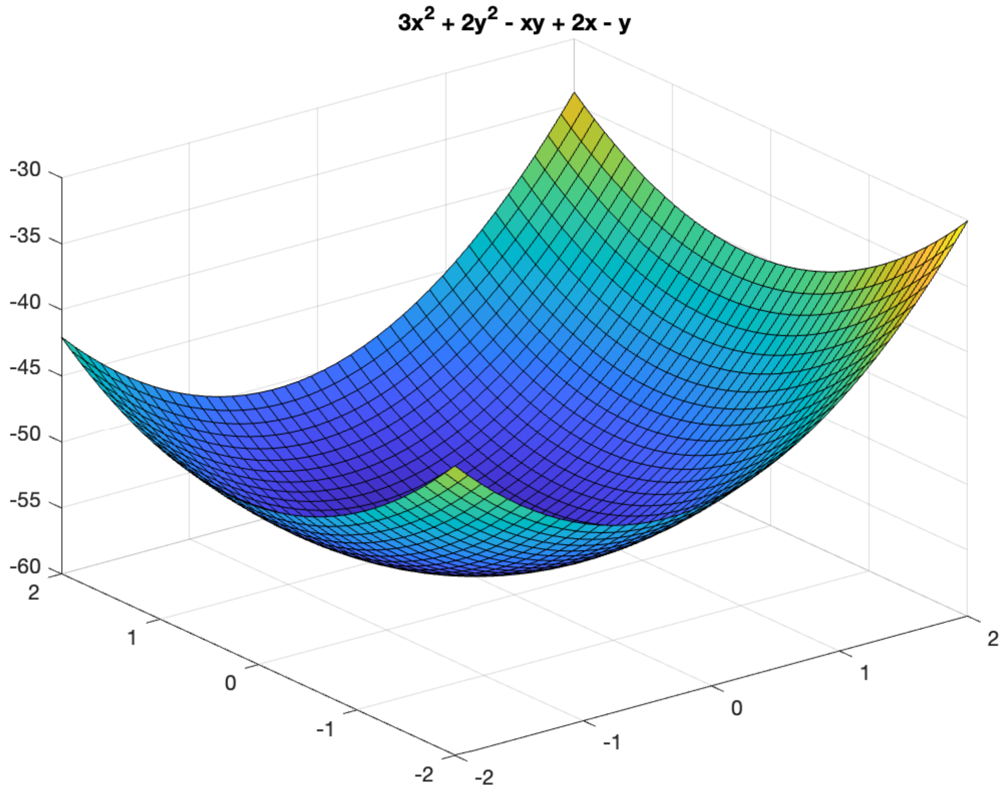
- Bivariate function:

$$f\colon \mathbb{R}^2 \longrightarrow \mathbb{R}, \qquad f(x, y) = 3x^2 - xy + 2y^2 + 2x - y$$

- Multivariate function: With $Q \in \mathbb{R}^{n,n}$ and $p \in \mathbb{R}^n$

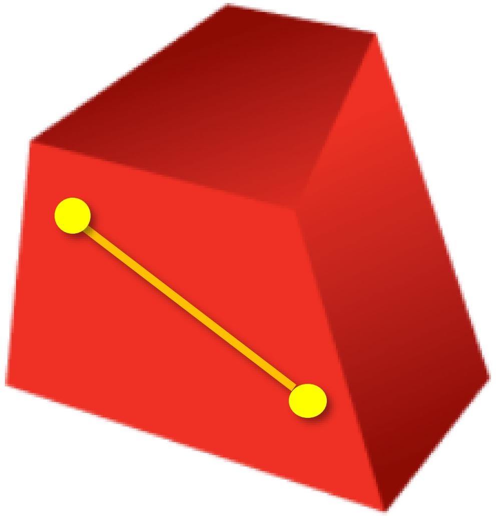
$$f\colon \mathbb{R}^n \rightarrow \mathbb{R}, \qquad f(x) = x^T Q\, x + p^T x$$


$x^2 + y^2$
8
7
6
5
4
3
2
1
0
2
1
0
-1
-2
-2
-1
0
1
2

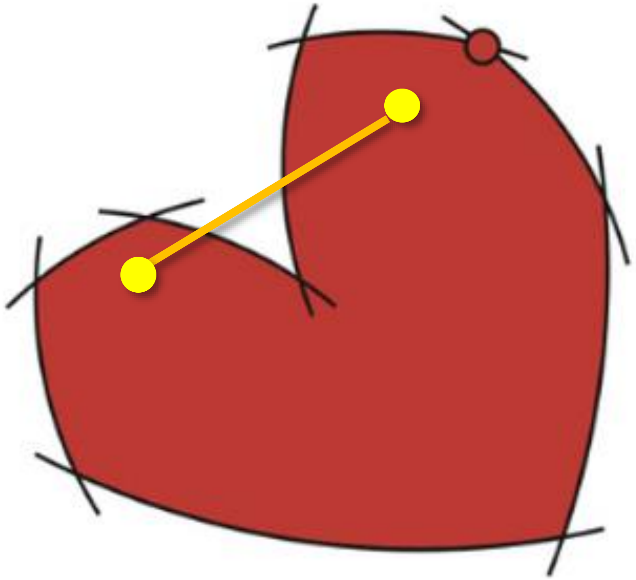

$3x^2 + 2y^2 - xy + 2x - y$
-30
-35
-40
-45
-50
-55
-60
2
1
0
-1
-2
-2
-1
0
1
2

Convex sets

A set in $\mathbb{R}^n$ is convex if any straight line connecting two points in the set is entirely contained in the set.



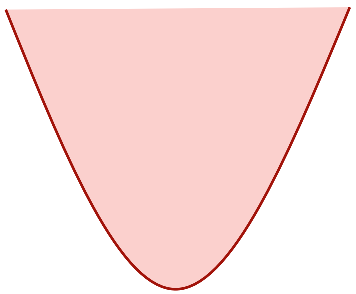
convex



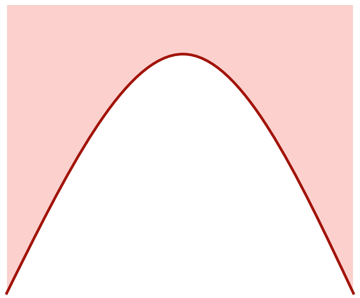
not convex

Convex functions and constraints

- A function f is convex if its epigraph $epi(f) := \{(x,t) | \ f(x) \leq t\} \subset \ \mathbb{R}^{n+1}$ is convex
- An inequality constraint $f(x) \leq 0$ is convex, if f is convex.



convex
$x^2 \leq t$



nonconvex
$-x^2 \leq t$

Every local optimum of a convex function is a global optimum!

Recognizing convex quadratic functions

Given some function $f(x) = \; x^T Q\, x + p^T x$, how can we decide whether f is convex?

- f is convex iff Q is positive semidefinite
- f is convex iff $x^T Q\, x \geq 0$ for all $x \in \mathbb{R}^n$

Important special cases

- $f(x) = (x_1 - a_1)^2 + (x_2 - a_2)^2 + \; \ldots + (x_n - a_n)^2$ (sum-of-squares)
- Q admits a matrix factorization $Q = F^T F$

Until further notice all quadratic functions in this lecture are assumed convex!

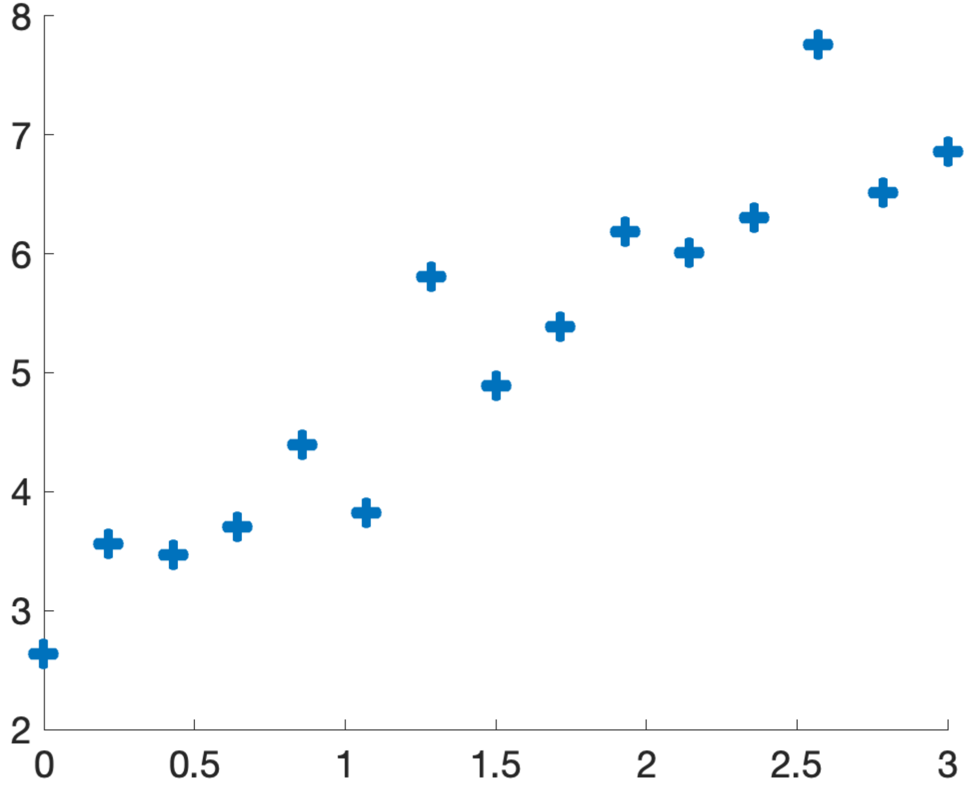
Problem formulation with a quadratic objective function

- Standard form of a Quadratic Program (QP):

$$\begin{array}{ll} min & x^T Q\, x + p^T x \\ s.t. & Ax = b \\ & x \geq 0 \end{array}$$

- Only difference: quadratic term in objective function
- (All kinds of linear inequality constraints allowed, “standard” form just normalizes the formulation).

Example: Linear regression



Goal: Find a straight line that "is close to all points"

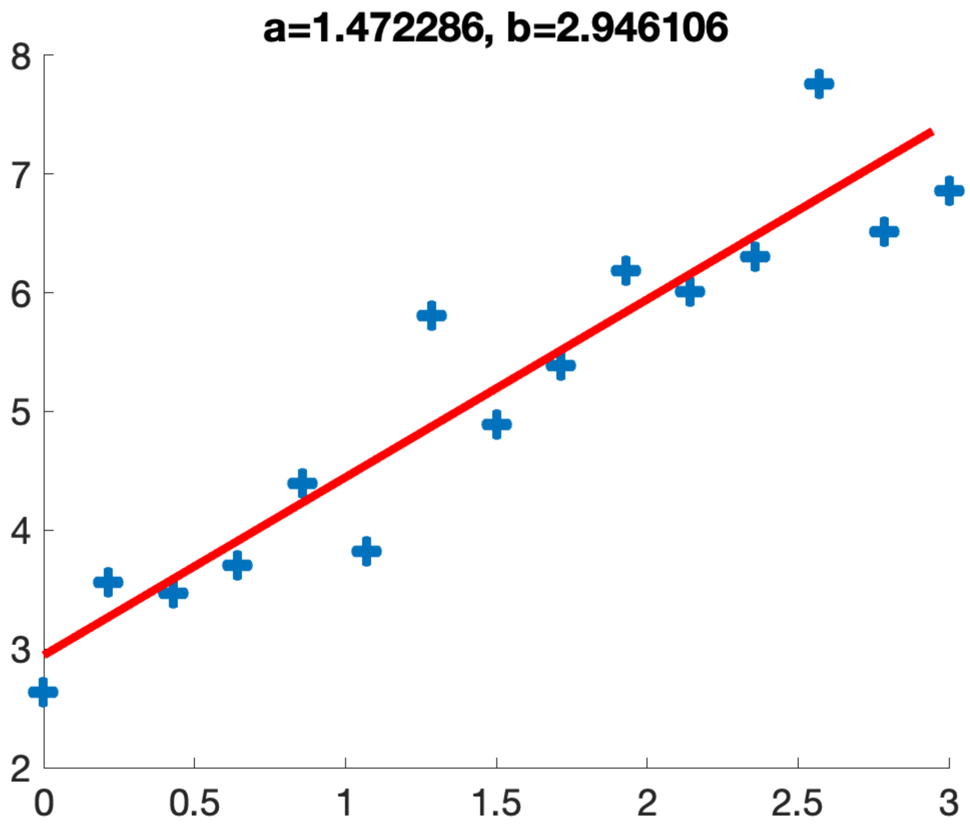
Linear regression II

- Linear model function: $ax + b$ with $a, b \geq 0$ (hyothetical physical meaning!)
- Data points: $(s_i, t_i) \in \mathbb{R}^2$
- Regression residual variables: $r_i = a * s_i + b - t_i$
- Fit error: $\|r\|_2 = \sqrt{r_1^2 + r_2^2 + \ldots + r_n^2}$

Putting it all in a QP:

$$\begin{array}{rc} min & r_1^2 + r_2^2 + \cdots + r_n^2 \\ s.t. & a * s_i + b - t_i - r_i = 0 \\ & a, b \geq 0 \end{array}$$

Linear regression III



Problem formulation with a quadratic constraint

- A Quadratically Constrained Program (QCP):

$$
\begin{array}{ll}
min & c^T x \\
s.t. & Ax = b \\
 & x^T Q\, x + p^T x \le 0 \\
 & x \ge 0
\end{array}
$$

- Here: One single quadratic constraint
- In general: Can have arbitrarily many such constraints
- In general: Can have a quadratic objective, too
- (All kinds of linear constraints allowed, "standard form" just for simplicity)

Example: Markowitz Portfolio model

- Assume the role of an investor
- We seek an "optimal" investment in assets
- There are n assets, and $0 \leq x_i \leq 1$ defines the fraction of our investment to be allocated to asset i
- Investments follow a stochastic model:
  - The return of the assets r is a random variable
  - Its mean $\mu = \mathbb{E}\mathrm{r}$, and covariance $\Sigma = \mathbb{E}(r - \mu)(r - \mu)^T$ are "known"
  - The return of the investment $y = r^T x$ has first and second moments $\mu^T x$ and $x^T \Sigma x$
- We want to maximize the expected return, while bounding the variance by a parameter γ:

$$\begin{array}{ll} max & \mu^T x \\ s.t. & x^T \Sigma x \leq \gamma \\ & x_1 + \cdots + x_n = 1 \\ & x \geq 0 \end{array}$$

Algorithms for QP and QCP

Recap: Algorithms for LP

- Simplex Algorithm
  - Exploits polyhedral structure of feasible region
  - Basic solutions correpond to vertices
  - One simplex iteration: Move from one vertex to an adjacent one
  - Typically takes “many” iterations, each single iteration typically very cheap. Sparse structure exploited in every algorithmic component
- Interior Point Method
  - Exploits analytic properties of the constraint functions
  - Iterates traverse the interior of the polyhedron
  - One interior point iteration: Move from “centered” point in the interior to the next
  - Typically takes “few” iterations, each single one is quite expensive. Sparse structure exploited for solving linear systems of equations.

Good News: Works for QP, too

- Both simplex and interior point methods extend quite naturally to quadratic objective functions
- Feasible region not structurally different: Still a polyhedron!
- But optimality conditions have more geometry now
- Consequence: Pivoting becomes more complicated, and gives more sources of numeric trouble

Gurobi comes with…

- Primal QP simplex algorithm (produces “basic” solutions)
- Dual QP simplex algorithm (produces “basic” solutions)
- Barrier algorithm for QP


```
Thread count: 8 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 15 rows, 17 columns and 44 nonzeros
Model fingerprint: 0x157aa79a
Model has 15 quadratic objective terms
Coefficient statistics:
  Matrix range     [2e-01, 3e+00]
  Objective range  [0e+00, 0e+00]
  QObjective range [2e+00, 2e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [3e+00, 8e+00]
Presolve removed 1 rows and 1 columns
Presolve time: 0.00s
Presolved: 14 rows, 16 columns, 42 nonzeros
Presolved model has 15 quadratic objective terms
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    0.0000000e+00   0.000000e+00   0.000000e+00      0s
       0    4.2997056e+02   0.000000e+00   3.471309e+02      0s
       3    3.5625798e+00   0.000000e+00   0.000000e+00      0s

Solved in 3 iterations and 0.00 seconds (0.00 work units)
```

```
Presolved model has 15 quadratic objective terms
Ordering time: 0.00s

Barrier statistics:
 Dense cols : 2
 AA' NZ     : 2.800e+01
 Factor NZ  : 4.500e+01
 Factor Ops : 1.310e+02 (less than 1 second per iteration)
 Threads    : 1

                  Objective                Residual
Iter       Primal          Dual         Primal    Dual     Compl     Time
   0   4.30955767e+02  -1.54332297e+05   1.93e+03 1.00e+03  9.94e+05     0s
   1   1.00700784e+06  -1.04761020e+06   1.93e-03 1.00e-03  1.28e+05     0s
                           <<<< SNIP >>>>

Barrier solved model in 10 iterations and 0.00 seconds (0.00 work units)
Optimal objective 3.56257979e+00
```

Algorithms for QCP

Simplex algorithm not easily extensible to quadratic constraints

- Mostly because vertex-view on solutions no longer matches the underlying geometry of the feasible set
- More general class of algorithms: Active set methods. Many conceptual similarities to simplex algorithms
- As of now cannot compete with Interior point methods, although for *special problem classes* specialized active set methods can shine…
- Not implemented in Gurobi

Interior point method for QCP

- Interior point algorithms can be extended to QCP, a trick is needed
- IPM extend *naturally* to rotated SOC constraints like

$$x_1^2 + \cdots + x_n^2 \leq 2z, \qquad z \geq 0$$

- Need a transformation from QC to SOC:
  - Original constraint: $\frac{1}{2}x^T Q\, x + p^T x \leq 0$
  - Since Q is PSD, it admits a factorization $Q = R^T R$
  - Add variables $y \in \mathbb{R}^n, q \geq 0$, then add constraints

$$\begin{aligned} y^T y \quad &\leq 2q \\ Rx \quad &= y \\ q + p^T x \quad &= 0 \end{aligned}$$

Spotting a QCP in the solver log file

Optimize a model with 26 rows, 60 columns and 102 nonzeros

Model fingerprint: 0x6c3c13d2

Model has 5 quadratic constraints

Coefficient statistics:

  Matrix range   [2e-01, 5e+00]

  QMatrix range   [1e+00, 1e+00]

  Objective range [1e+00, 1e+01]

  Bounds range   [0e+00, 0e+00]

  RHS range   [1e+00, 1e+00]

Presolve removed 9 rows and 30 columns

Presolve time: 0.00s

Presolved: 17 rows, 30 columns, 65 nonzeros

Presolved model has 5 second-order cone constraints

Ordering time: 0.00s

Adding integrality constraints

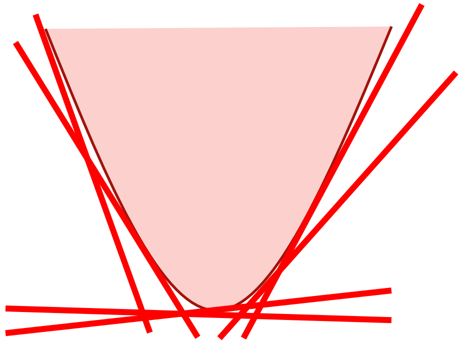
- MIQP: Some additional integrality and/or SOS constraints
  - All techniques from “Introduction to algorithms” apply
  - Workhorse: QP simplex
  - Interesting specialized techniques come into play, too
- MIQCP:
  - More challanging
  - More algorithmic machinery needed

Outer approximation for MIQCP

- *In principle* MIQCP fits naturally into the B&B framework
  - Relaxing integrality constraints yields convex subproblem (QCP!)
  - Branching on the fractional integers effects the same implicit enumeration as for MILP
- But many tricks from MILP do not carry over directly
  - Solution of the QCP subproblem at each tree node is expensive (relative to a re-solve with simplex).
  - Heuristics that rely on efficient simplex warm start cannot be run
  - Cutting planes that need a Simplex basis matrix cannot be deduced
  - …
- Sometimes all of this doesn't do much harm overall, but for many models these drawbacks are severe
- Another trick is needed

MIQCP outer approximation

- Assume for simplicity that we have a MILP with one additional standard SOC constraint
$$f(x) = -x_1^2 + x_2^2 + \ldots + x_n^2 \leq 0,$$
- Next simply forget that the SOC constraint exist, and solve "just" the LP relaxation of the resulting MILP, call the optimal solution vector x^*
- If $f(x^*) \leq 0$: We are done!
- Otherwise find a separating hyperplane for x^*
- Add as new linear constraint to the LP
- Resolve and repeat!



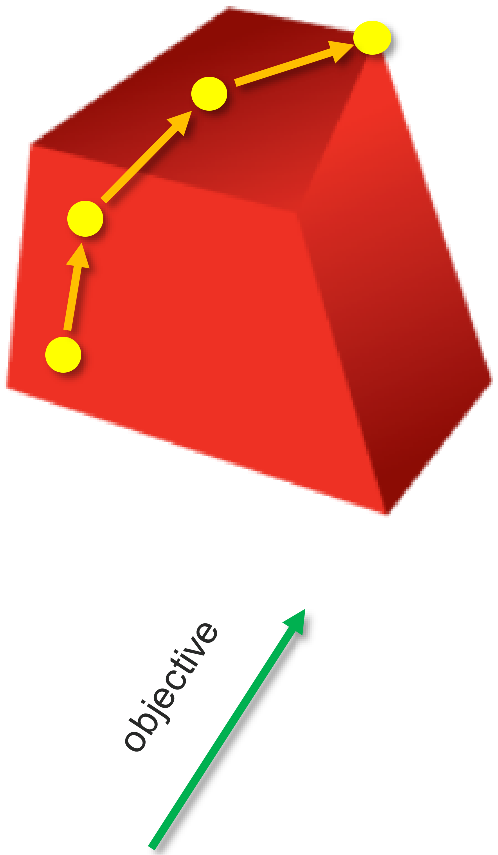
Unfortunately it's even more complicated

- Often we need further transformations to solve a model efficiently
- The overall strategy is controlled by parameters "PreMIQCPForm" and "MIQPCmethod"
- Happy to explain more during the coffee break!

From convex to nonconvex

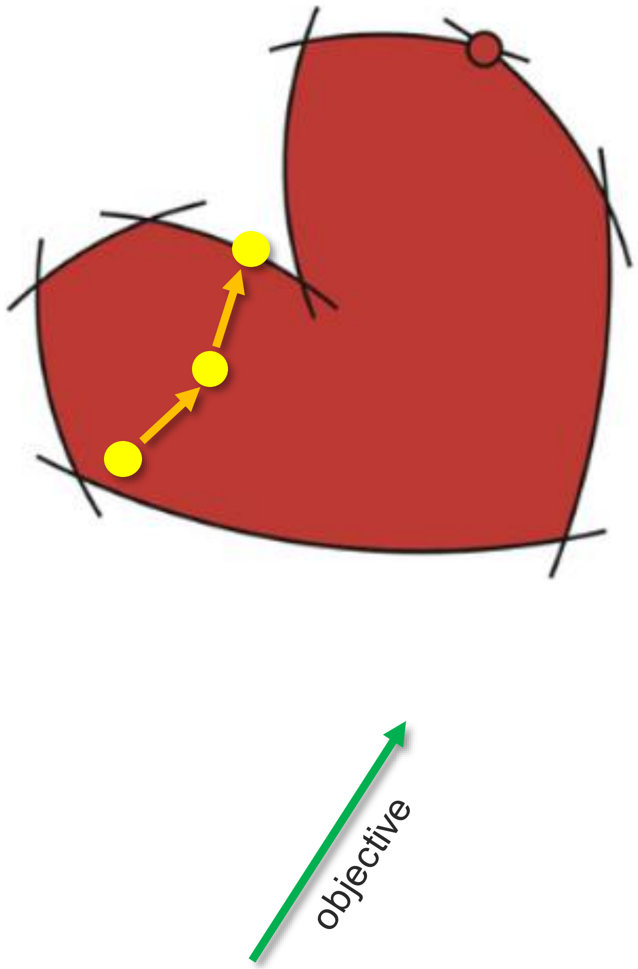
Convex optimization is „easy“

- How „difficult“ is it to optimize a smooth convex function over a convex domain?
- Hypothetical gradient-descend like algorithm:
  - Start at some feasible point
  - Find direction along which the objective function decreaes
  - Take step into that direction without leaving the domain
  - Convexity implies: With only very mild conditions on the step lengths, we always converge to globally optimal solution!!!
- Theory and practice for convex optimization well developed
  - Strong convergence results
  - Efficient algorithms



Nonconvexity and optimization

- How „difficult“ is it to optimize a smooth convex function over a nonconvex domain?
- Hypothetical gradient-descend like algorithm:
    - Start at some feasible point
    - Find direction along which the objective function decreaes
    - Take step into that direction without leaving the domain
    - No matter what we do, we can only guarantee to converge to a locally optimal solution
- This problem is in fact NP-hard!
- Optimizing a smooth, nonconvex function over a convex set is NP-hard, too!



Convex and nonconvex Optimization

Linear Objective
$\min c^T x$

Quadratic Objective
$\min a^T x + x^T Q x$
with PSD Q

SOS Constraints

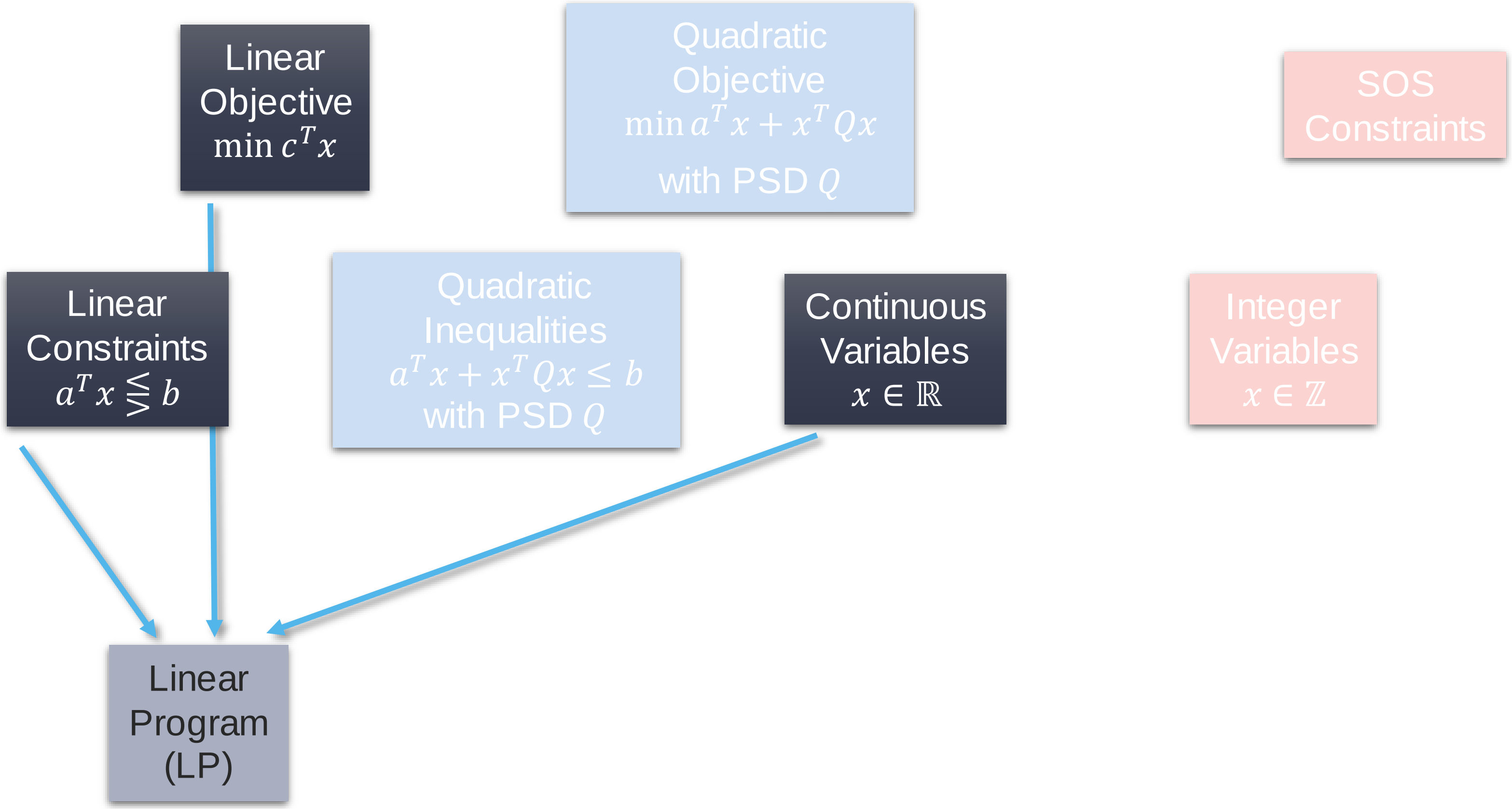
Linear Constraints
$a^T x \lesseqqgtr b$

Quadratic Inequalities
$a^T x + x^T Q x \leq b$
with PSD Q

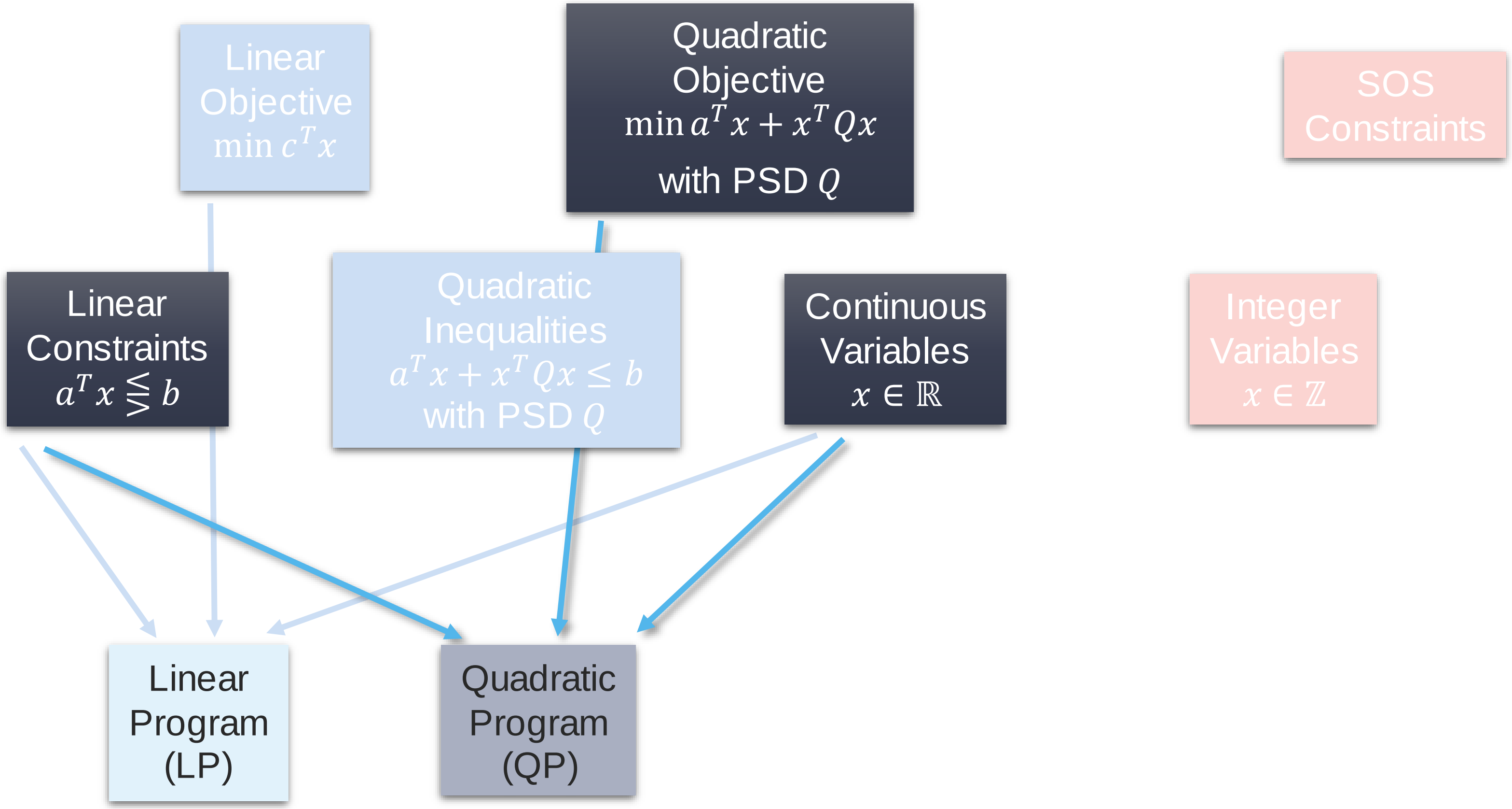
Continuous Variables
$x \in \mathbb{R}$

Integer Variables
$x \in \mathbb{Z}$

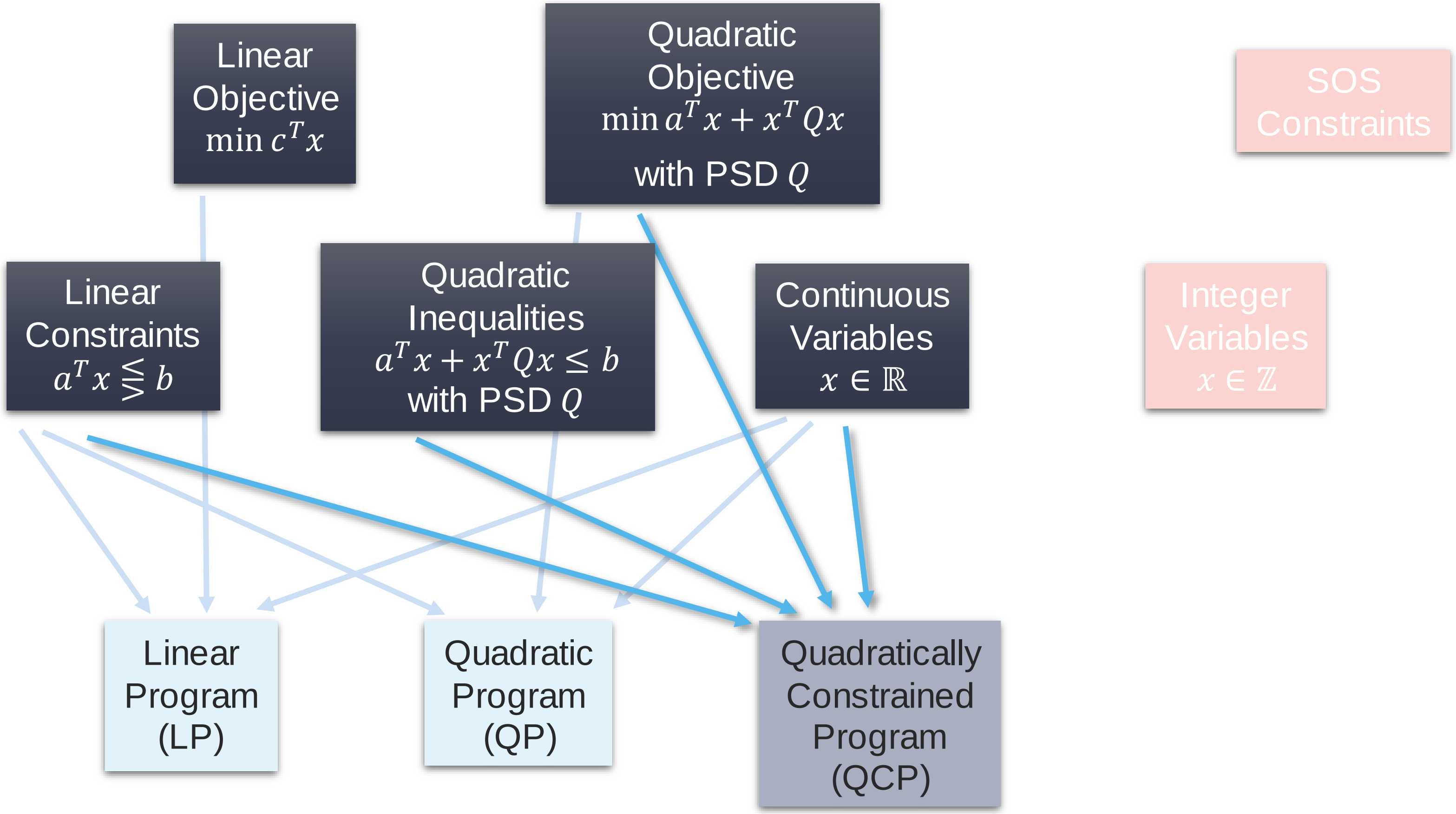
Convex and nonconvex Optimization



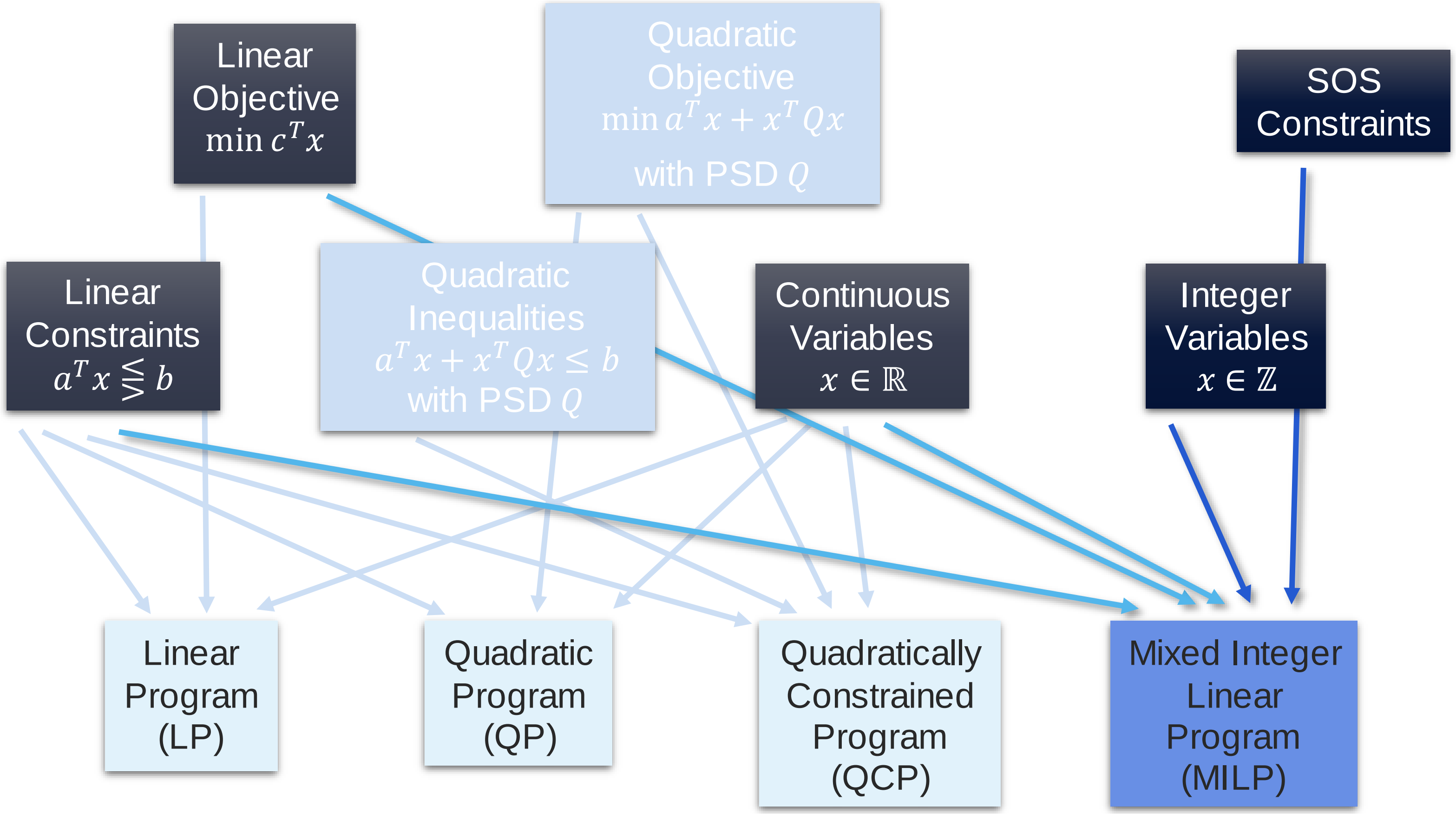
Convex and nonconvex Optimization



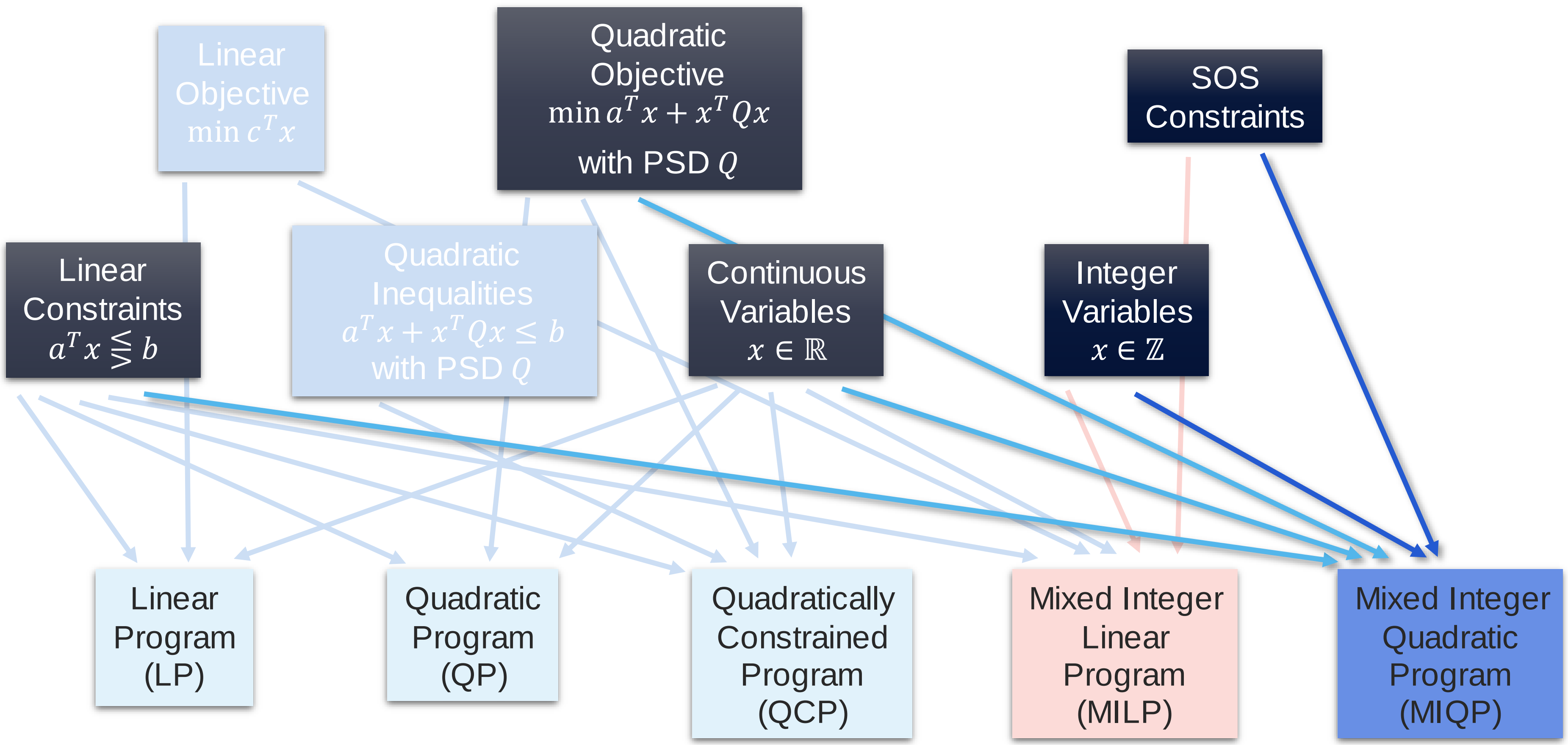
Convex and nonconvex Optimization



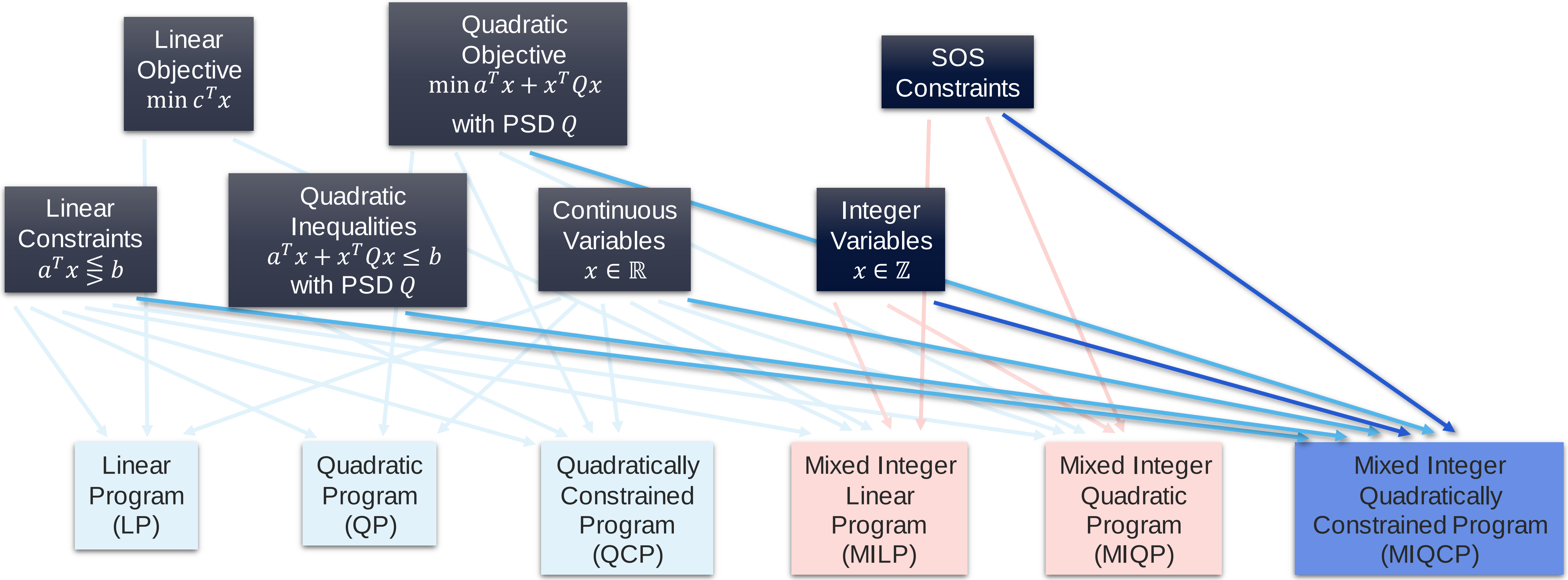
Convex and nonconvex Optimization



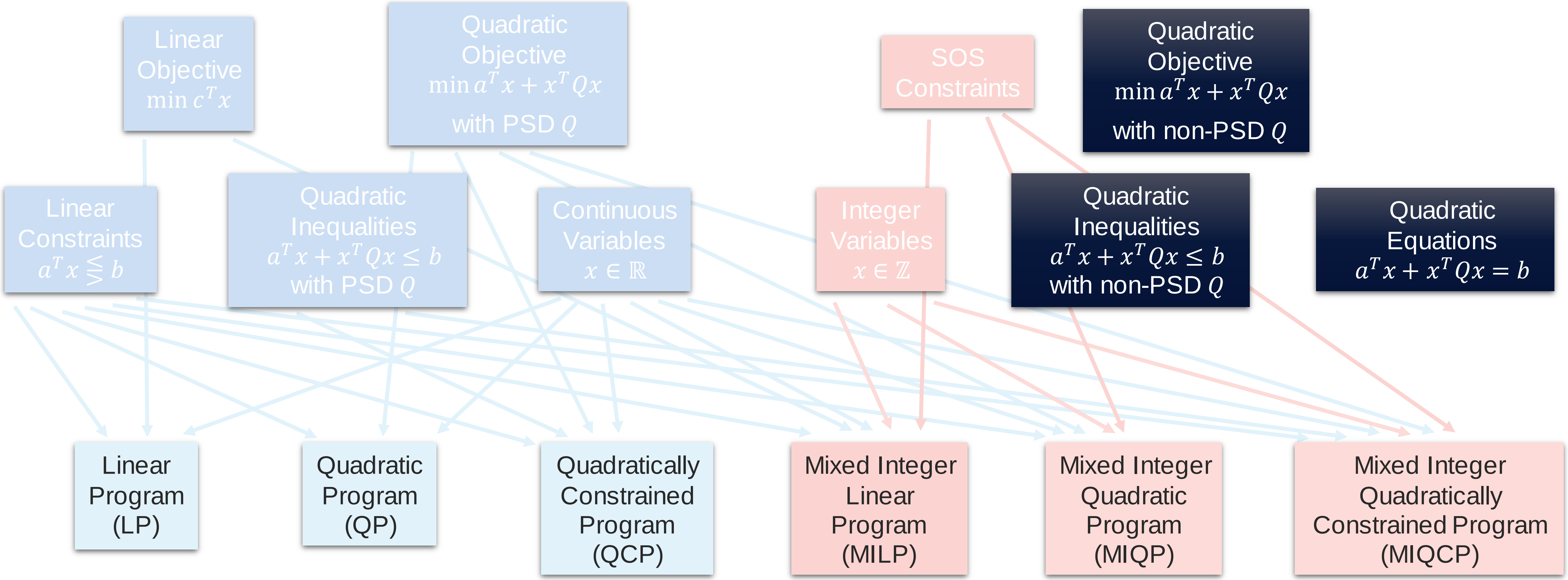
Convex and nonconvex Optimization



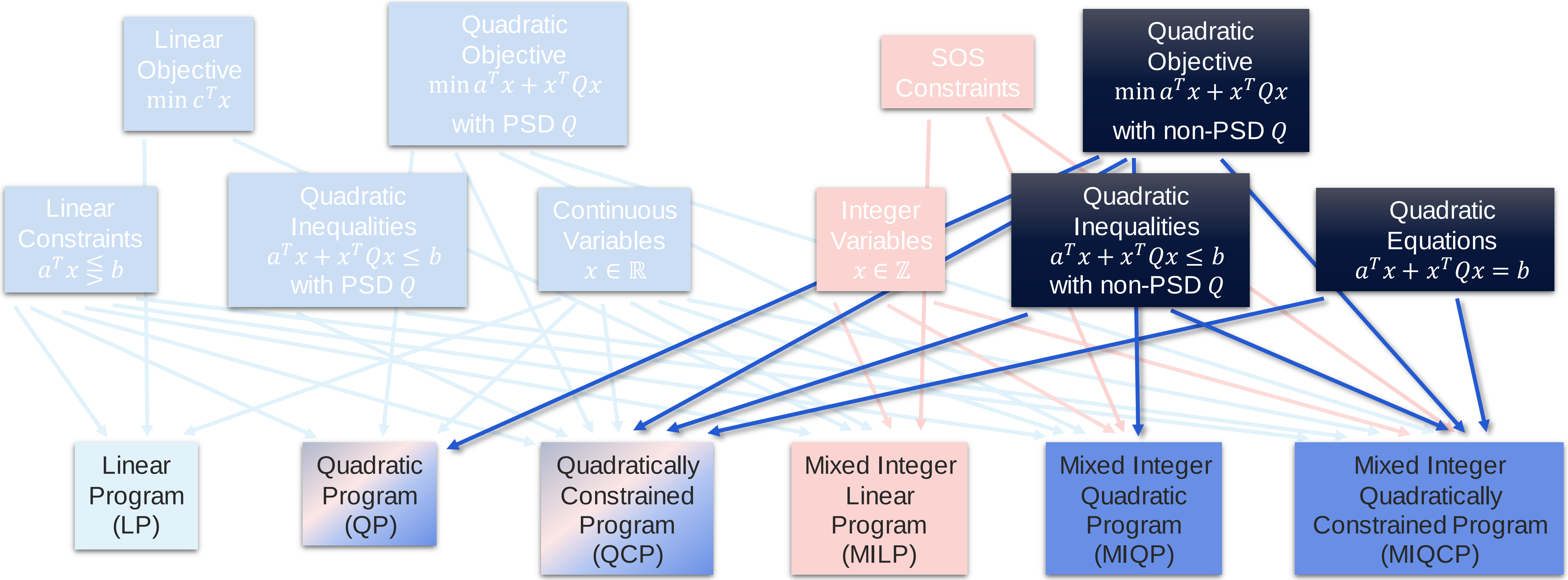
Convex and nonconvex Optimization



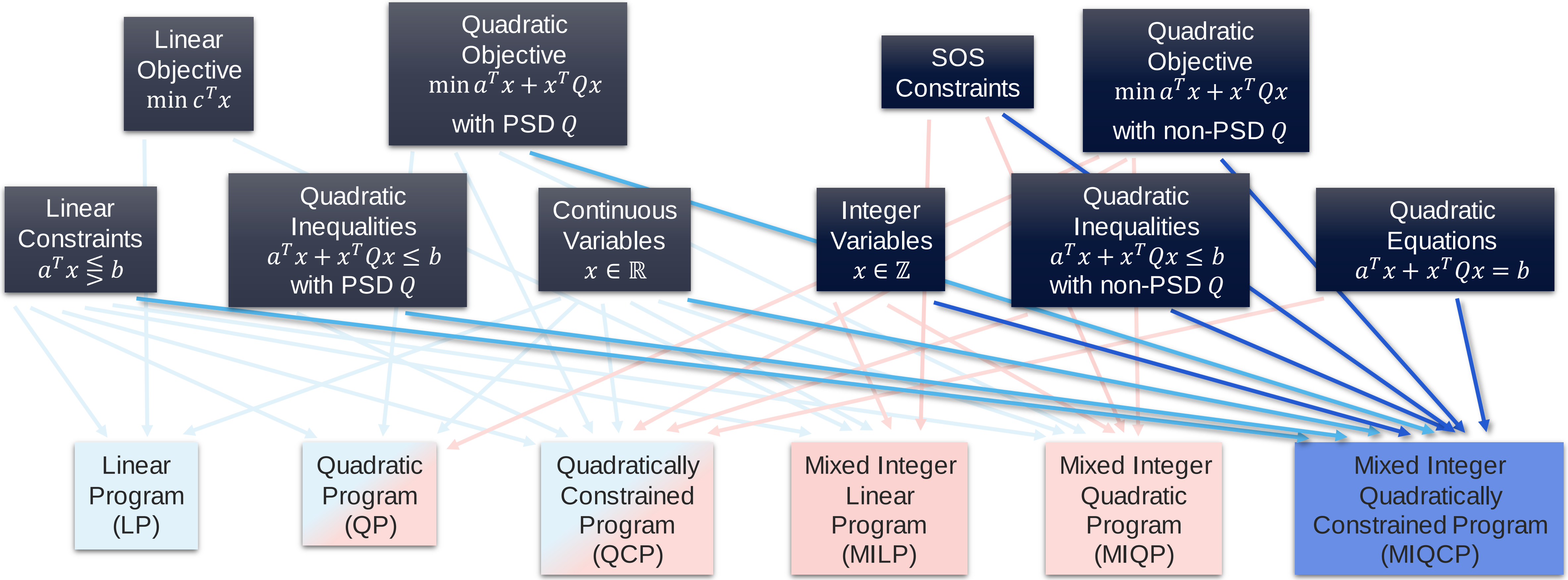
Convex and nonconvex Optimization



Convex and nonconvex Optimization



Convex and nonconvex Optimization



Mixed Integer Quadratically Constrained Programming

- A Mixed Integer Quadratically Constrained Program (MIQCP) is defined as

$$
\begin{array}{lccccccl}
\min & c^T x & + & x^T Q_0 x & & & \\
\text{s.t.} & a_1^T x & + & x^T Q_1 x & \leq & b_1 & \\
 & & \cdots & & & & \\
 & a_m^T x & + & x^T Q_m x & \leq & b_m & \\
 & l & \leq & x & \leq & u & \\
 & & & x_j & \in & \mathbb{Z} & \text{for all } j \in I
\end{array}
$$

- The Q_k are symmetric matrices
- If all Q_k are positive semi-definite, then QCP relaxation is convex
- What if quadratic constraints or objective are non-convex?

Nonconvex QP, QCP, MIQP, and MIQCP

GUROBI OPTIMIZATION

- Applications
  - Pooling problem (blending problem is LP, pooling introduces intermediate pools → bilinear)
  - Petrochemical industry (oil refinery: constraints on ratio of components in tanks)
  - Wastewater treatment
  - Emissions regulation
  - Agricultural / food industry (blending based on pre-mix products)
  - Mining
  - Energy
  - Production planning (constraints on ratio between internal and external workforce)
  - Logistics (restrictions from free trade agreements)
  - Water distribution (Darcy-Weisbach equation for volumetric flow)
  - Engineering design
  - Finance (constraints on exchange rates)
- General MINLP
  - Non-convex MIQCP can model polynomial problems of arbitrary degree
  - Solve general MINLPs by approximating as polynomial problem
    - but: will often fail for higher degrees due to numerical issues

Nonconvex QP, QCP, MIQP, and MIQCP

- Traditional nonconvex constraints: Integer variables, SOS constraints
- Since version 9.0: Bilinear constraints: $z = xy$
  - Allows one to represent arbitrary nonconvex quadratic inequalities and equations
- These nonconvexities are treated by
  - Cutting planes
  - Branching

- Translation of nonconvex quadratic constraints into bilinear constraints:

$3x_1^2 - 7x_1x_2 + 2x_1x_3 - x_2^2 + 3x_2x_3 - 5x_3^2 = 12$ (nonconvex quad. constraint)

$z_{11} \coloneqq x_1^2, z_{12} \coloneqq x_1x_2, z_{13} \coloneqq x_1x_3, z_{22} \coloneqq x_2^2, z_{23} \coloneqq x_2x_3, z_{33} \coloneqq x_3^2$ (6 bilinear constraints)

$3z_{11} - 7z_{12} + 2z_{13} - z_{22} + 3z_{23} - 5z_{33} = 12$ (linear constraint)

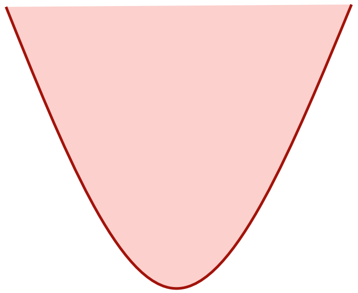
More Details on Bilinear Transformation

- Special cases to avoid bilinear constraint for $q_{ij}x_ix_j$ term
  - At least one of x_i and x_j is fixed
  - Square of a binary: $x_i^2 = x_i$
  - At least one of x_i and x_j is binary: $z_{ij} := x_ix_j$ can easily be modeled
    - if possible, add big-M linearization for $z_{ij} := x_ix_j$
    - otherwise, add SOS1 formulation for $z_{ij} := x_ix_j$
  - Square term $q_{ii}x_i^2$ with $q_{ii} > 0$: term is convex
- For quadratic inequalities $a^Tx + x^TQx \leq b$, only one side of $z_{ij} := x_ix_j$ is needed
  - $z_{ij} \geq x_ix_j$, if $q_{ij} > 0$
  - $z_{ij} \leq x_ix_j$, if $q_{ij} < 0$
- More sophisticated partitions into convex and nonconvex parts are possible and may work better!

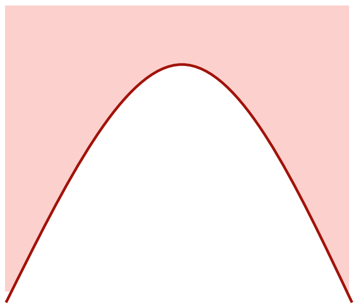
Dealing With Bilinear Constraints

- General form: $a^T z + dxy \leqq b$ (linear sum plus single product term, inequality or equation)

- Consider square case ($x = y$):



convex
$-z + x^2 \leq 0$

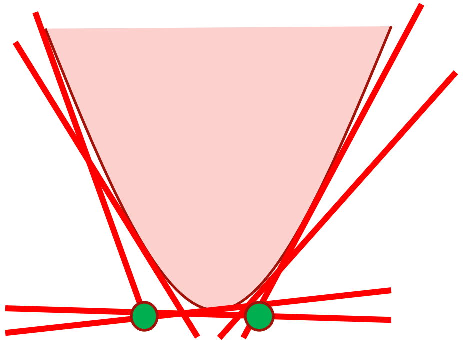


nonconvex
$-z - x^2 \leq 0$

Dealing With Bilinear Constraints

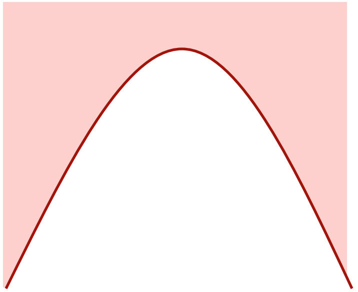
- General form: $a^T z + dxy \leqq b$ (linear sum plus single product term, inequality or equation)

- Consider square case ($x = y$):



convex
$-z + x^2 \leq 0$

easy: add tangent cuts



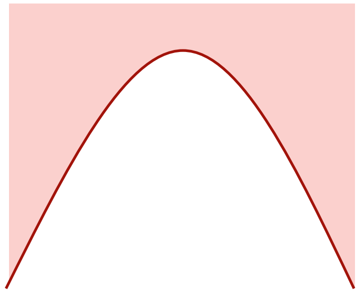
nonconvex
$-z - x^2 \leq 0$

Dealing With Bilinear Constraints

- General form: $a^T z + dxy \leqq b$ (linear sum plus single product term, inequality or equation)

- Consider square case ($x = y$):

nonconvex
$-z - x^2 \leq 0$

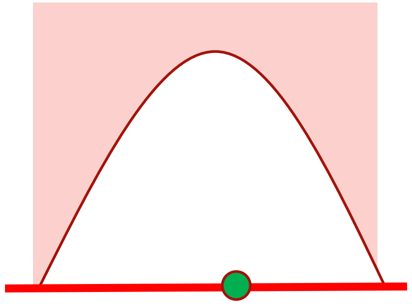


Dealing With Bilinear Constraints

- General form: $a^T z + dxy \leqq b$ (linear sum plus single product term, inequality or equation)

- Consider square case $(x = y)$:

nonconvex
$-z - x^2 \leq 0$



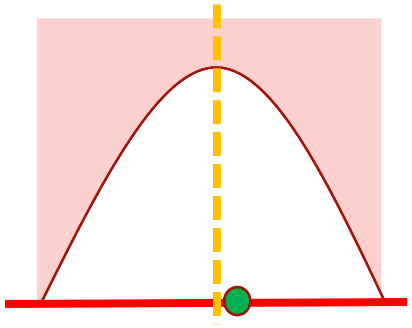
Dealing With Bilinear Constraints

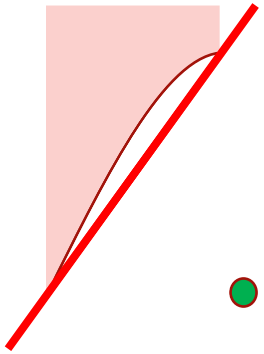
- General form: $a^T z + dxy \leqq b$ (linear sum plus single product term, inequality or equation)

- Consider square case $(x = y)$:

nonconvex

$-z - x^2 \leq 0$

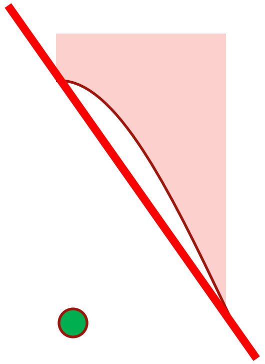




branching

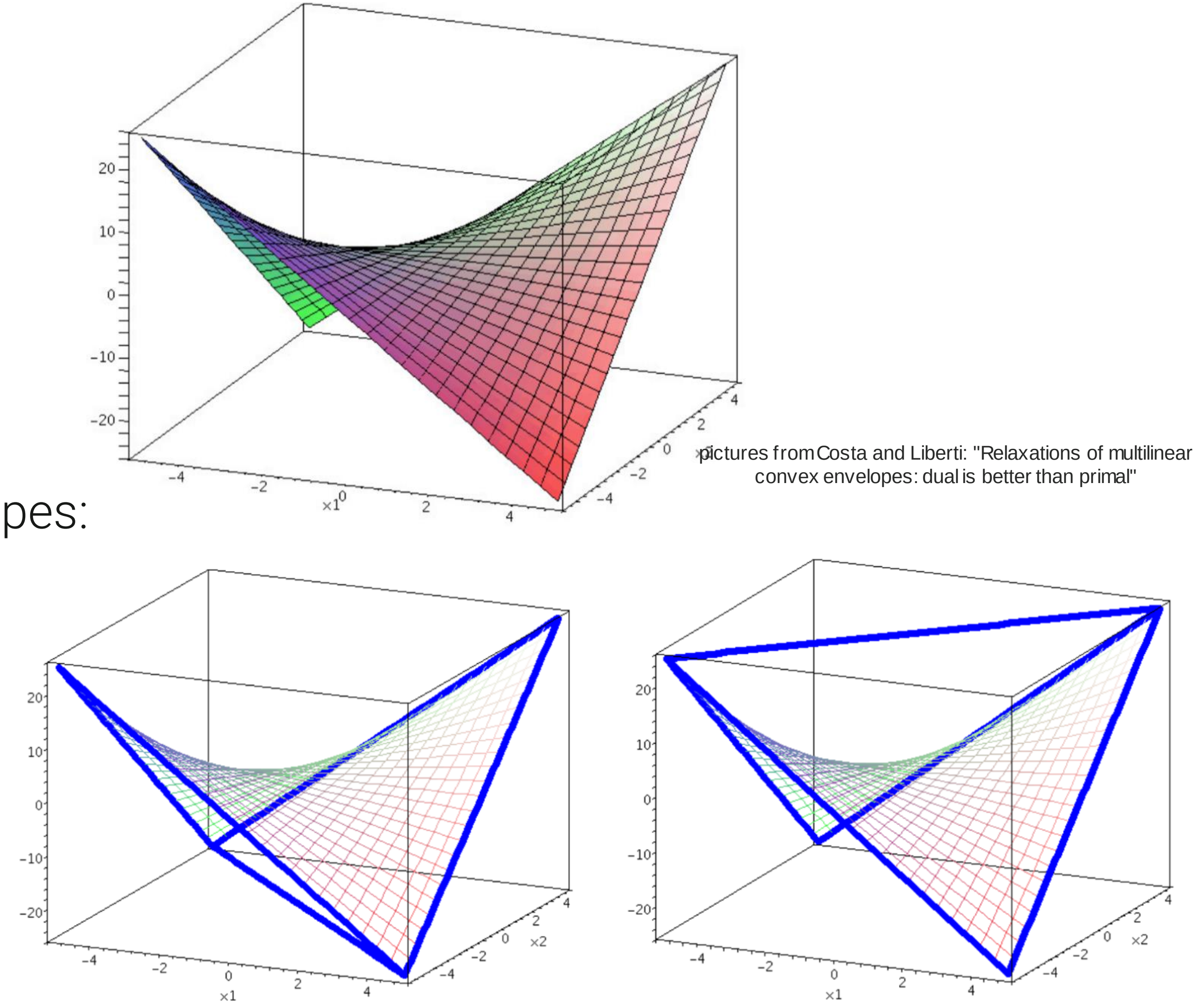
$x \leq 0$ or $x \geq 0$

update relaxation locally



LP Relaxation of Bilinear Constraints

Mixed product case: $-z + xy = 0$



pictures from Costa and Liberti: "Relaxations of multilinear convex envelopes: dual is better than primal"

McCormick lower and upper envelopes:

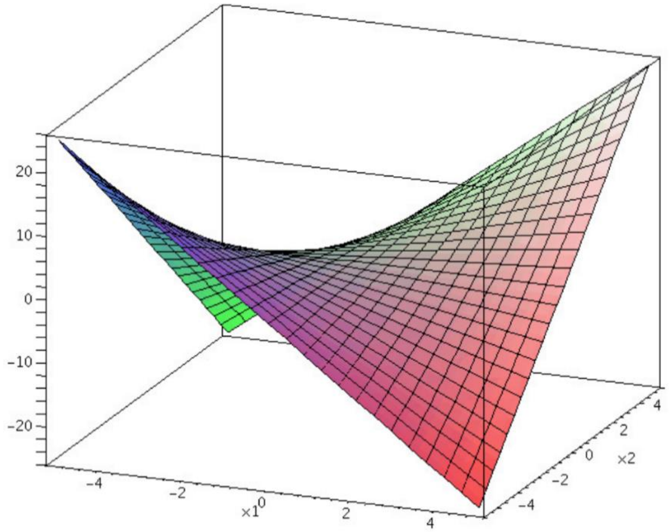
$$-z + l_x y + l_y x \leq l_x l_y$$
$$-z + u_x y + u_y x \leq u_x u_y$$

$$-z + u_x y + l_y x \geq u_x l_y$$
$$-z + l_x y + u_y x \geq l_x u_y$$

GUROBI
OPTIMIZATION

LP Relaxation of Bilinear Constraints

Mixed product case: $-z + xy = 0$



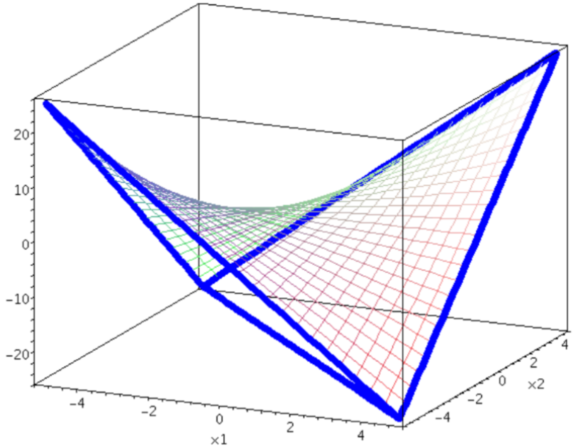
pictures from Costa and Liberti: "Relaxations of multilinear convex envelopes: dual is better than primal"

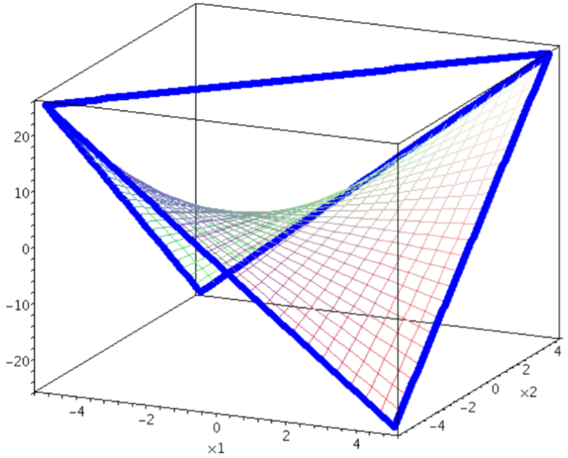
McCormick lower and upper envelopes:

$$-z + l_x y + l_y x \leq l_x l_y$$
$$-z + u_x y + u_y x \leq u_x u_y$$

$$-z + u_x y + l_y x \geq u_x l_y$$
$$-z + l_x y + u_y x \geq l_x u_y$$

coefficients depend on local bounds





Adaptive Constraints in LP Relaxation

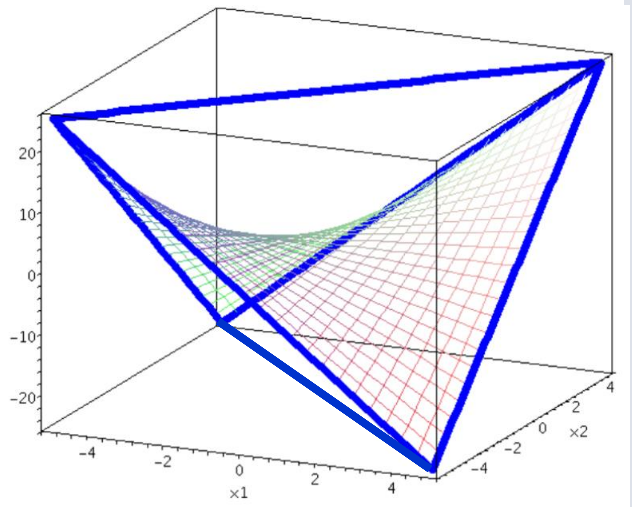
- Coefficients and right hand sides of McCormick constraints depend on local bounds of variables
  - Whenever local bounds change, LP coefficients and right hand sides are updated
  - May lead to singular or ill-conditioned basis
    - in worst case, simplex needs to start from scratch

- Alternative to adaptive constraints: locally valid cuts
  - Add tighter McCormick relaxation on top of weaker, more global one, to local node
  - Advantages:
    - old simplex basis stays valid in all cases
      - more global McCormick constraints will likely become slack and basic
    - should lead to fewer simplex iterations
  - Disadvantages:
    - basis size (number of rows) changes all the time during solve
      - complicated (and potentially time and memory consuming) data management needed
    - redundant more global McCormick constraints stay in LP
      - LP solver performs useless calculations in linear system solves

Spatial Branching

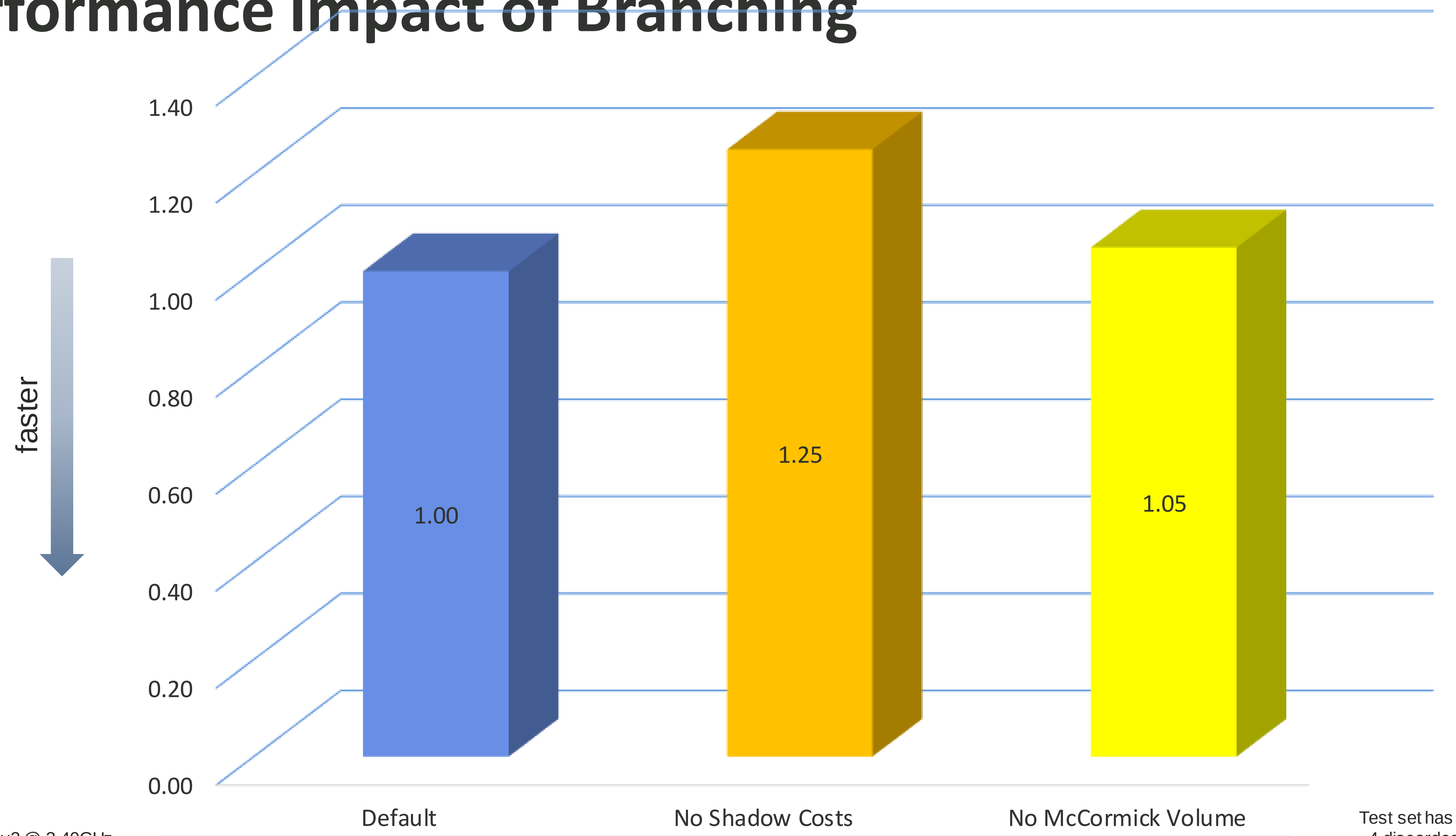
- Branching variable selection
  - What most solvers do: first branching on fractional integer variables as usual
  - If no fractional integer variable exists, select continuous variable in violated bilinear constraint
  - Our variable selection rule is a combination of:
    - sum of absolute bilinear constraint violations
    - reduce McCormick volume as much as possible
      - big McCormick polyhedron is turned into two smaller McCormick polyhedra after branchi
      - sum of smaller volumes is smaller than big volume
    - shadow costs of variable for linear constraints

- Branching value selection
  - We use a standard way
    - a convex combination of LP value and mid point of current domain
  - Avoid numerical pitfalls
    - large branching values for unbounded variables
    - tiny child domains if LP value is very close to bound
    - very deep dives (node selection)



$z_{ij} = x_i\, x_j$

Performance Impact of Branching



Timeouts:	5	60	8

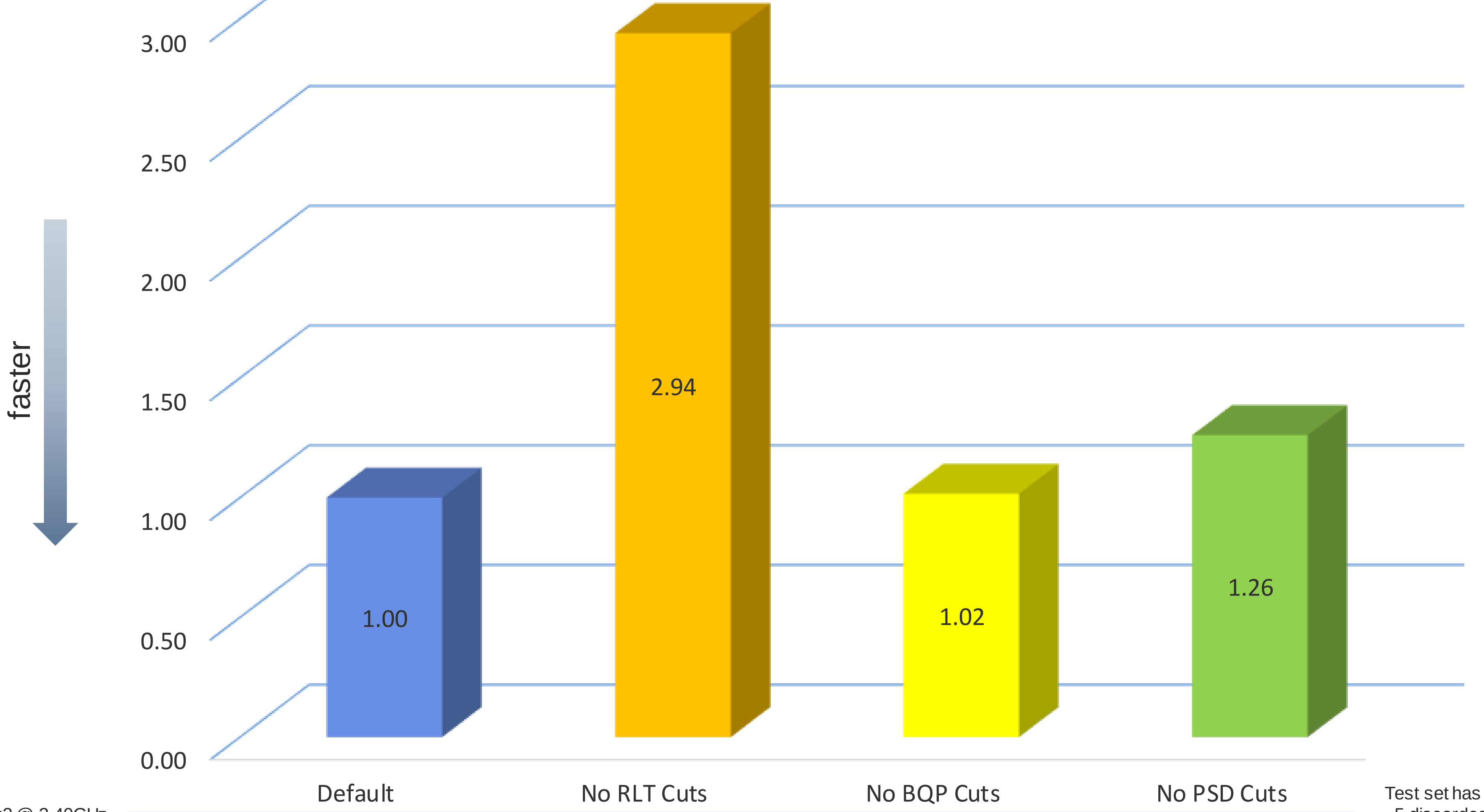
Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v3 @ 3.40GHz
4 cores, 8 hyper-threads
32 GB RAM

Test set has 444 models, using 5 random seeds:
- 4 discarded due to inconsistent answers
- 54 discarded that none of the versions can solve
- speed-up measured on >10s bracket: 143 models

Cutting Planes for Mixed Bilinear Programs

- All MILP cutting planes apply
- Special cuts for bilinear constraints
  - RLT Cuts
    - Reformulation Linearization Technique (Sherali and Adams, 1990)
    - multiply linear constraints with single variable, linearize resulting product terms
    - very powerful for bilinear programs, also helps a bit for convex MIQCPs and MILPs
  - BQP Cuts
    - facets from Boolean Quadric Polytope (Padberg 1989)
      - equivalent to Cut Polytope
    - currently implemented: Padberg's clique cuts for BQP
  - PSD Cuts
    - tangents of PSD cone defined by $Z = xx^T$ relationship: $Z - xx^T \succcurlyeq 0$ (Sherali and Fraticelli, 2002)

Performance Impact of Cutting Planes



Timeouts:	2	114	7	13

Time limit: 10000 sec.
Intel Xeon CPU E3-1240 v3 @ 3.40GHz
4 cores, 8 hyper-threads
32 GB RAM

Test set has 477 models, using 5 random seeds:
- 5 discarded due to inconsistent answ ers
- 55 discarded that none of the versions can solve
- speed-up measured on >10s bracket: 158 models

Summary

- Quadratic optimization if fun
- Convex quadratic optimization is an established technique, and Gurobi has a broad toolset for it
- Nonconvex quadratic optimization is mathematical challanging discipline, Gurobi puts a lot of effort into making it tractable in practice
- Check out my other lecture “Nonconvex optimization under the hood” talk in the advanced track! (*Caution: contains math*)

