On affine conic and disjunctive constraints in the upcoming MOSEK version 10

INFORMS Annual Meeting 2021
October 24th, Anaheim, CA

Sven Wiese
(joint work with Erling D. Andersen)

www.mosek.com

MOSEK

A software package/library for solving:

- Linear and conic problems.
- Convex quadratically constrained problems.
- Also mixed-integer versions of the above.

Current version: 9.3.

Conic optimization in standard form:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array} \qquad \begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & c - A^T y \in (\mathcal{K})^* \end{array}$$

$\mathcal{K}$ is a nonempty pointed convex cone, $(\mathcal{K})^*$ its dual.

MOSEK

A software package/library for solving:

- Linear and conic problems.
- Convex quadratically constrained problems.
- Also mixed-integer versions of the above.

Current version: 9.3.

Conic optimization in standard form:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array} \qquad \begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & c - A^T y \in (\mathcal{K})^* \end{array}$$

$\mathcal{K}$ is a nonempty pointed convex cone, $(\mathcal{K})^*$ its dual.

The cones available in **MOSEK**

- the nonnegative orthant

$$\mathbb{R}^n_+ := \{x \in \mathbb{R}^n \mid x_j \geq 0,\ j = 1, \ldots, n\}$$

- the quadratic cone

$$\mathcal{Q}^n = \{x \in \mathbb{R}^n \mid x_1 \geq \left(x_2^2 + \cdots + x_n^2\right)^{1/2} = \|x_{2:n}\|_2\}$$

- the rotated quadratic cone

$$\mathcal{Q}^n_r = \{x \in \mathbb{R}^n \mid 2x_1x_2 \geq x_3^2 + \cdots + x_n^2 = \|x_{3:n}\|_2^2,\ x_1, x_2 \geq 0\}$$

- the semidefinite matrix cone

$$\mathcal{S}^n = \{X \in \mathbb{R}^{n\times n} \mid z^T X z \geq 0,\ \forall z \in \mathbb{R}^n\}$$

The cones available in **MOSEK**

- the nonnegative orthant

$$\mathbb{R}^n_+ := \{x \in \mathbb{R}^n \mid x_j \geq 0,\ j = 1, \ldots, n\}$$

- the quadratic cone

$$\mathcal{Q}^n = \{x \in \mathbb{R}^n \mid x_1 \geq \left(x_2^2 + \cdots + x_n^2\right)^{1/2} = \|x_{2:n}\|_2\}$$

- the rotated quadratic cone

$$\mathcal{Q}^n_r = \{x \in \mathbb{R}^n \mid 2x_1x_2 \geq x_3^2 + \cdots + x_n^2 = \|x_{3:n}\|_2^2,\ x_1, x_2 \geq 0\}$$

- the semidefinite matrix cone

$$\mathcal{S}^n = \{X \in \mathbb{R}^{n\times n} \mid z^T X z \geq 0,\ \forall z \in \mathbb{R}^n\}$$

The cones available in **MOSEK**

- the nonnegative orthant

$$\mathbb{R}^n_+ := \{x \in \mathbb{R}^n \mid x_j \geq 0,\ j = 1, \ldots, n\}$$

- the quadratic cone

$$\mathcal{Q}^n = \{x \in \mathbb{R}^n \mid x_1 \geq \left(x_2^2 + \cdots + x_n^2\right)^{1/2} = \|x_{2:n}\|_2\}$$

- the rotated quadratic cone

$$\mathcal{Q}_r^n = \{x \in \mathbb{R}^n \mid 2x_1x_2 \geq x_3^2 + \cdots + x_n^2 = \|x_{3:n}\|_2^2,\ x_1, x_2 \geq 0\}$$

- the semidefinite matrix cone

$$\mathcal{S}^n = \{X \in \mathbb{R}^{n \times n} \mid z^T X z \geq 0,\ \forall z \in \mathbb{R}^n\}$$

The cones available in **MOSEK**

- the nonnegative orthant

$$\mathbb{R}^n_+ := \{x \in \mathbb{R}^n \mid x_j \geq 0,\ j = 1, \ldots, n\}$$

- the quadratic cone

$$\mathcal{Q}^n = \{x \in \mathbb{R}^n \mid x_1 \geq \left(x_2^2 + \cdots + x_n^2\right)^{1/2} = \|x_{2:n}\|_2\}$$

- the rotated quadratic cone

$$\mathcal{Q}_r^n = \{x \in \mathbb{R}^n \mid 2x_1x_2 \geq x_3^2 + \cdots + x_n^2 = \|x_{3:n}\|_2^2,\ x_1, x_2 \geq 0\}$$

- the semidefinite matrix cone

$$\mathcal{S}^n = \{X \in \mathbb{R}^{n\times n} \mid z^T X z \geq 0,\ \forall z \in \mathbb{R}^n\}$$

The cones available in **MOSEK**

MOSEK also supports some non-symmetric cones:

- the three-dimensional exponential cone

$$\mathcal{K}_{exp} = \mathrm{cl}\{x \in \mathbb{R}^3 \mid x_1 \geq x_2 \exp(x_3/x_2),\ x_2 > 0\}.$$

- the three-dimensional power cone

$$\mathcal{P}^{\alpha} = \{x \in \mathbb{R}^3 \mid x_1^{\alpha} x_2^{(1-\alpha)} \geq |x_3|,\ x_1, x_2 \geq 0\},$$

for $0 < \alpha < 1$.

These 5 cones together are highly versatile for convex modeling!

The cones available in **MOSEK**

MOSEK also supports some non-symmetric cones:

- the three-dimensional exponential cone

$$\mathcal{K}_{exp} = \mathrm{cl}\{x \in \mathbb{R}^3 \mid x_1 \geq x_2 \exp(x_3/x_2),\ x_2 > 0\}.$$

- the three-dimensional power cone

$$\mathcal{P}^\alpha = \{x \in \mathbb{R}^3 \mid x_1^\alpha x_2^{(1-\alpha)} \geq |x_3|,\ x_1, x_2 \geq 0\},$$

for $0 < \alpha < 1$.

These 5 cones together are highly versatile for convex modeling!

The cones available in **MOSEK**

MOSEK also supports some non-symmetric cones:

- the three-dimensional exponential cone

$$\mathcal{K}_{exp} = \mathrm{cl}\{x \in \mathbb{R}^3 \mid x_1 \geq x_2 \exp(x_3/x_2),\ x_2 > 0\}.$$

- the three-dimensional power cone

$$\mathcal{P}^{\alpha} = \{x \in \mathbb{R}^3 \mid x_1^{\alpha} x_2^{(1-\alpha)} \geq |x_3|,\ x_1, x_2 \geq 0\},$$

for $0 < \alpha < 1$.

These 5 cones together are highly versatile for convex modeling!

The cones available in **MOSEK**

MOSEK also supports some non-symmetric cones:

- the three-dimensional exponential cone

$$\mathcal{K}_{exp} = \mathrm{cl}\{x \in \mathbb{R}^3 \mid x_1 \geq x_2 \exp(x_3/x_2),\ x_2 > 0\}.$$

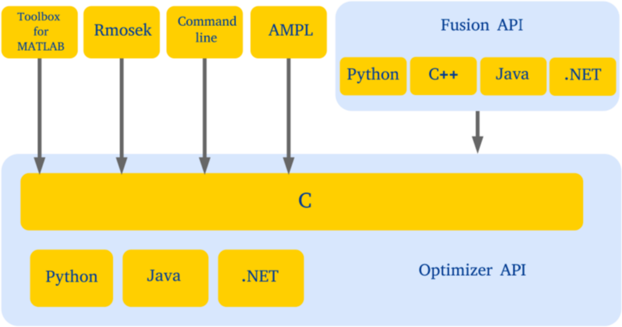
- the three-dimensional power cone

$$\mathcal{P}^{\alpha} = \{x \in \mathbb{R}^3 \mid x_1^{\alpha} x_2^{(1-\alpha)} \geq |x_3|,\ x_1, x_2 \geq 0\},$$

for $0 < \alpha < 1$.

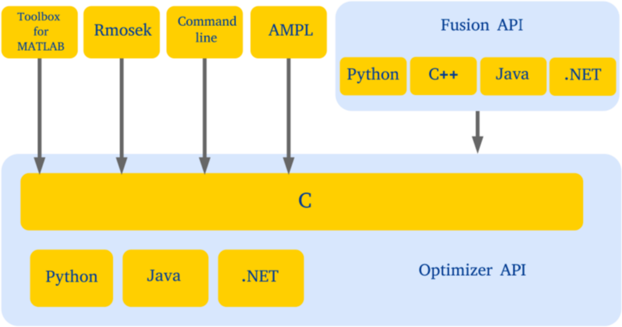
These 5 cones together are highly versatile for convex modeling!

Interfaces to **MOSEK**



The C API (this talk) is the core of **MOSEK** and all other APIs are built on top of that.

Interfaces to **MOSEK**



The C API (this talk) is the core of **MOSEK** and all other APIs are built on top of that.

Challenge: (Re-)Design an interface

MOSEK version 9:

$$(x_5; x_6; x_8) \in \mathcal{Q}^3$$

- In principle general: Use artificial variables.
- Example: $t \geq \|Wx - d\|_2 \iff (t; s) \in \mathcal{Q}^{p+1}, s = Wx - d$.
- Cumbersome to use at least for some models.
- Want to easily specify an affine expression of the variables belonging to a cone.

Challenge: (Re-)Design an interface

MOSEK version 9:

$$(x_5; x_6; x_8) \in \mathcal{Q}^3$$

- In principle general: Use artificial variables.
- Example: $t \geq \|Wx - d\|_2 \Longleftrightarrow (t; s) \in \mathcal{Q}^{p+1}, s = Wx - d.$
- Cumbersome to use at least for some models.
- Want to easily specify an affine expression of the variables belonging to a cone.

Challenge: (Re-)Design an interface

Hence, handle the constraint type

$$F^k x + f^k \in \mathcal{K}^k + g^k.$$

- How to build an interface for this type of constraint?
- That works in a low level language like C?
- Efficient, i.e. low space and computational overhead?
- Extensible to new cone types?

Why the g? Reason: reuse of affine expressions:

$$2 \leq x + y + 1 \text{ and } x + y + 1 \leq 6.$$

Challenge: (Re-)Design an interface

Hence, handle the constraint type

$$F^k x + f^k \in \mathcal{K}^k + g^k.$$

- How to build an interface for this type of constraint?
- That works in a low level language like C?
- Efficient, i.e. low space and computational overhead?
- Extensible to new cone types?

Why the g? Reason: reuse of affine expressions:

$$2 \leq x + y + 1 \text{ and } x + y + 1 \leq 6.$$

Challenge: (Re-)Design an interface

Hence, handle the constraint type

$$F^k x + f^k \in \mathcal{K}^k + g^k.$$

- How to build an interface for this type of constraint?
- That works in a low level language like C?
- Efficient, i.e. low space and computational overhead?
- Extensible to new cone types?

Why the g? Reason: reuse of affine expressions:

$$2 \leq x + y + 1 \text{ and } x + y + 1 \leq 6.$$

New: the affine expressions storage

Define

$$Fx + f,$$

a storage/dictionary of affine expressions.

- F is sparse matrix and f is a dense vector.
- Affine expressions can be appended but never deleted.
- Variables can be appended to and deleted from x.
- Affine expressions can be modified.

New: the affine expressions storage

Define

$$Fx + f,$$

a storage/dictionary of affine expressions.

- F is sparse matrix and f is a dense vector.
- Affine expressions can be appended but never deleted.
- Variables can be appended to and deleted from x.
- Affine expressions can be modified.

New: the affine expressions storage

With that,

$$F^k x + f^k = F_{\mathcal{I},:} x + f_{\mathcal{I}}$$

where $\mathcal{I}$ is an ordered list of indexes.

- F^k and f^k are not provided explicitly.
- Represented by a list of indexes into the affine expression storage.

New: the affine expressions storage

With that,

$$F^k x + f^k = F_{\mathcal{I},:} x + f_{\mathcal{I}}$$

where $\mathcal{I}$ is an ordered list of indexes.

- F^k and f^k are not provided explicitly.
- Represented by a list of indexes into the affine expression storage.

Also new: the domains storage

Introduce

$$\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2, \ldots),$$

a list of domains.

A domain

- Has a dimension d.
- Has a type, e.g. the exponential cone type.
- Has potentially some associated parameters, e.g. α for the power cone.
- Can never be deleted.

Also new: the domains storage

Introduce

$$\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2, \ldots),$$

a list of domains.

A domain

- Has a dimension d.
- Has a type, e.g. the exponential cone type.
- Has potentially some associated parameters, e.g. α for the power cone.
- Can never be deleted.

An affine conic constraint

An affine conic constraint in **MOSEK** 10 consists of:

- An ordered list of affine expressions indexes.
- A domain index k.
- A g vector.

and represents

$$F_{\mathcal{I},:}x + f_{\mathcal{I}} \in \mathcal{D}_k + g.$$

- Affine conic constraints can be appended, deleted and modified.
- Dimension checking is easy.
- Easily extensible with new cone types.
- Everything is easily implementable in C.
- Can be implemented efficiently.

An affine conic constraint

An affine conic constraint in **MOSEK** 10 consists of:

- An ordered list of affine expressions indexes.
- A domain index k.
- A g vector.

and represents

$$F_{\mathcal{I},:}x + f_{\mathcal{I}} \in \mathcal{D}_k + g.$$

- Affine conic constraints can be appended, deleted and modified.
- Dimension checking is easy.
- Easily extensible with new cone types.
- Everything is easily implementable in C.
- Can be implemented efficiently.

Another novelty: disjunctive constraints

A disjunctive constraint is of the form

$$[D^1 x \leq b^1] \vee [D^2 x \leq b^2] \vee \ldots \vee [D^l x \leq b^l].$$

- Can model semi-continuous variables: $[x = 0] \vee [l \leq x \leq u]$.
- Complementarity constraints: $s \cdot t = 0 \iff [s = 0] \vee [t = 0]$.
- Piecewise linear functions:

  $$[f = x, 0 \leq x \leq 1] \vee [f = 1 - 2x, 1 \leq x \leq 2] \vee [f = -3, 2 \leq x].$$

- Indicator constraints: $[z = 0] \vee \left[d^T x \leq b\right]$.
- Many more...

Another novelty: disjunctive constraints

A disjunctive constraint is of the form

$$[D^1 x \leq b^1] \vee [D^2 x \leq b^2] \vee \ldots \vee [D^l x \leq b^l].$$

- Can model semi-continuous variables: $[x = 0] \vee [l \leq x \leq u]$.
- Complementarity constraints: $s \cdot t = 0 \iff [s = 0] \vee [t = 0]$.
- Piecewise linear functions:

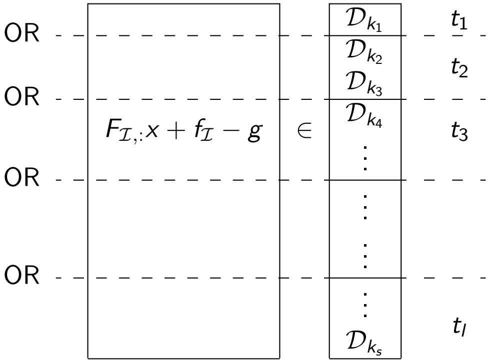
$$[f = x, 0 \leq x \leq 1] \vee [f = 1 - 2x, 1 \leq x \leq 2] \vee [f = -3, 2 \leq x].$$

- Indicator constraints: $[z = 0] \vee \left[d^T x \leq b\right]$.
- Many more...

Use the affine expressions and domain storages!

A disjunctive constraint in **MOSEK** 10 consists of:

- An ordered list of affine expression indexes.
- An ordered list of domain indexes (only linear ones for now!).
- A g vector.
- A list of term sizes $t_1, \ldots, t_l$:



Summary

- Described how to specify affine conic constraints in **MOSEK** 10.
  - Efficient yet quite general when implemented in C.
  - Easy to extend to new cone types.
- Described how to specify disjunctive constraints in **MOSEK** 10.
  - Generalize many special constructs such semi-continuous variables, indicator constraints, etc.
  - Make it possible to get rid of big-Ms.
- Documentation at `mosek.com/documentation/`
  - Modeling cook book / cheat sheet.
  - Manuals for interfaces.
  - Notebook collection.
- Tutorials and more at `github.com/MOSEK/`

Summary

- Described how to specify affine conic constraints in **MOSEK** 10.
  - Efficient yet quite general when implemented in C.
  - Easy to extend to new cone types.

- Described how to specify disjunctive constraints in **MOSEK** 10.
  - Generalize many special constructs such semi-continuous variables, indicator constraints, etc.
  - Make it possible to get rid of big-Ms.

- Documentation at `mosek.com/documentation/`
  - Modeling cook book / cheat sheet.
  - Manuals for interfaces.
  - Notebook collection.
- Tutorials and more at `github.com/MOSEK/`

Summary

- Described how to specify affine conic constraints in **MOSEK** 10.
  - Efficient yet quite general when implemented in C.
  - Easy to extend to new cone types.

- Described how to specify disjunctive constraints in **MOSEK** 10.
  - Generalize many special constructs such semi-continuous variables, indicator constraints, etc.
  - Make it possible to get rid of big-Ms.

- Documentation at `mosek.com/documentation/`
  - Modeling cook book / cheat sheet.
  - Manuals for interfaces.
  - Notebook collection.
- Tutorials and more at `github.com/MOSEK/`