The Conic Benchmark Format

Version 4

Technical Reference Manual
November 2, 2021

Document of the Conic Benchmark Library

Abstract

This document constitutes the technical reference manual of the Conic Benchmark Format (version 4) with file extension `.cbf` or `.CBF`. This is a file storage format consolidating problem instances of optimization with mixed-integer variables over mixed-conic linear, second-order, semidefinite, exponential and power cone domains. The format has been designed with benchmark libraries in mind, and therefore focuses on compact and easily parsable representations. The problem structure is separated from data, and the format moreover facilitate benchmarking of hotstart capability through a sequence of incremental changes.

Acknowledgements

Henrik A. Friberg is the creator of the CBF format and the author of this document. The format is inspired by ideas presented by Imre Pólik. Version 1 came to life under the valuable feedback of Erling D. Andersen and Mathias Stolpe and supports real-valued symmetric cones. Version 2 defines the exponential cone in response to requests by Chris Coey and Miles Lubin. Version 3 adds syntax for parametric cones and defines the radial power cone. Version 4 defines the one norm cone, infinity norm cone, a vectorized semidefinite cone and the radial geometric mean cone (a common non-parametric specialization of the radial power cone), along with simplified counterparts as suggested by Lea Kapelevich; the hypograph power cone and geometric mean hypograph cone.

Funding

This document was funded by MOSEK ApS. First version co-funded with the Danish Ministry of Science, Innovation and Higher Education.

Contents

1 Introduction

The Conic Benchmark Format (version 4) is backwards compatible with all previous versions. The readings of an old parser able to process a recent CBF-v4 file without error, are thus valid if the parser is made well-enough to check and emit errors on unknown/unexpected input. In terms of content, the format encodes optimization problem instances of the form

$$\begin{array}{ll} \text{minimize or maximize} & g^{obj} \\ \text{subject to} & g_1, \ldots, g_m \text{ in conic domains,} \\ & G_1, \ldots, G_M \text{ in conic domains,} \end{array}$$

where g^{obj} and g_i (resp. G_i) are scalar (resp. matrix) valued affine expressions of scalar and matrix variables, x and X, that may themselves be constrained to conic domains. Scalar variables may further be restricted individually to take integer values only. The format thus encode mixed-integer conic optimization problems mixing primal and dual standard form.

1.1 Minimal working example

The conic optimization problem (1), has three variables in a quadratic cone Q^3 - first one is integer - and an affine expression in domain $\{0\}$ (equality constraint).

$$\begin{array}{ll} \text{minimize} & 5.1\, x_0 \\ \text{subject to} & 6.2\, x_1 + 7.3\, x_2 - 8.4 \in \{0\} \\ & x \in Q^3,\ x_0 \in \mathbb{Z}. \end{array} \tag{1}$$

Its formulation in the Conic Benchmark Format begins with the version of the CBF format used, to safeguard against later revisions.

```
VER
4
```

Next follows the problem structure, consisting of the objective sense, the number and domain of variables, the indices of integer variables, and the number and domain of scalar-valued affine expressions (i.e., the equality constraint).

```
OBJSENSE
MIN

VAR
3 1
Q 3

INT
1
0

CON
1 1
L= 1
```

Finally follows the problem data, consisting of the coefficients of the objective, the coefficients of the constraints, and the constant terms of the constraints. All data is specified on a sparse coordinate form.

```
OBJACOORD
1
0 5.1

ACOORD
2
0 1 6.2
0 2 7.3

BCOORD
1
0 -8.4
```

This concludes the example! Please see Section 2 and Section 3 for details about the document structure and the use of keywords. More examples found in Appendix C.

2 The structure of CBF files

This section defines how information is written in the CBF format, without being specific about the type of information being communicated.

2.1 Information blocks

The format is composed as a list of information blocks. The first line of an information block is the `KEYWORD`, revealing the type of information provided. The second line - of some keywords only - is the `HEADER`, revealing any unknown dimensions of information provided, thereby enabling parsers to preallocate data structures. The remaining lines are the `BODY` holding the actual information.

```
KEYWORD
BODY

KEYWORD
HEADER
BODY
```

The `KEYWORD` specifies how each line in the `HEADER` and `BODY` is structured. In fact, the exact number of lines in the `BODY` is decidable either from the `KEYWORD`, the `HEADER`, or from another information block required to precede it. Under some keywords the `BODY` is subdivided into chunks consisting of separate `CHUNKHEADER` and `CHUNKBODY` elements.

2.2 Information order by group

All information blocks belong to exactly one of four groups of information. These information groups, and the order they must appear in, is:

1. File format.
2. Parametric cone specification.
3. Problem structure.
4. Problem data.

The first group, *file format*, provides information on how to interpret the file. It is currently limited to the keyword `VER`, specifying the version of the CBF format in use. The second group, *parametric cone specification*, define lookup tables of parametric cones for specific parameter settings. The third group, *problem structure*, gives the information needed to deduce the size and variable/constraint types of the problem instance. Finally, the fourth group, *problem data*, specifies the coefficients and constants of the problem instance.

2.3 Embedded hotstart-sequences

A sequence of problem instances, based on the same problem structure, is allowed within a single file. This is facilitated via the `CHANGE` keyword used within the problem data information group, as a separator between the information blocks of each instance. The information blocks following a `CHANGE` keyword is appending to, or changing (e.g., setting coefficients back to their default value of zero), the problem data of the preceding instance.

The sequence is intended for benchmarking of hotstart capability, where the solvers can reuse their internal state and solution (subject to the achieved accuracy) as warmpoint for the succeeding instance. Whenever this feature is unsupported or undesired, the keyword `CHANGE` may be interpreted as the end of file.

2.4 File encoding and line width restrictions

The format is based on the US-ASCII printable character set with two extensions as listed below. Note, by definition, that none of these extensions can be misinterpreted as printable US-ASCII characters:

- A line feed marks the end of a line, carriage returns are ignored.
- Comment-lines may contain unicode characters in UTF-8 encoding.

The line width is restricted to 512 bytes, with 3 bytes reserved for the potential carriage return, line feed and null-terminator.

Integers and floating point numbers must follow the ISO C decimal string representation in the standard "C" locale. The format does not impose restrictions on the magnitude of, or number of significant digits in, numeric data, but the use of 64-bit integers and 64-bit IEEE 754 floating point numbers should be sufficient to avoid loss of precision.

2.5 Comment-line and whitespace rules

The format allows single-line comments respecting the following rule:

- Lines having first byte equal to '`#`' (US-ASCII 35) are comments, and should be ignored. Comments are only allowed between information blocks.

Given that a line is not a comment-line, whitespace characters should be handled according to the following rules:

- Leading and trailing whitespace characters should be ignored.
- The separator between multiple pieces of information on one line, is either one or more whitespace characters.
- Lines containing only whitespace characters are empty, and should be ignored. Empty lines are only allowed between information blocks.

3 How instances are specified

Conic optimization problems consist of variables, constraints and one objective function. In the CBF format, these are defined as follows.

- Variables are either scalar-valued (part of a vector restricted to a cone), or matrix-valued (restricted to be symmetric positive semidefinite). These are referred to as the scalar variables, x_j for $j \in \mathcal{J}$, and PSD variables, X_j for $j \in \mathcal{J}^{PSD}$. Only scalar variables can be integer.
- Constraints are affine expressions of the variables, either scalar-valued (part of a vector restricted to a cone), or matrix-valued (restricted to be symmetric positive semidefinite). These are thus referred to as the scalar constraints, with affine expressions g_i for $i \in \mathcal{I}$, and PSD constraints, with affine expressions G_i for $i \in \mathcal{I}^{PSD}$.
- The objective is a scalar-valued affine expression of the variables, either to be minimized or maximized. We refer to this expression as g^{obj}.

3.1 Cones and their parametric specification

The format uses an explicit syntax for symmetric positive semidefinite cones as shown in Section 3.2. For scalar variables and constraints, constructed in vectors, the supported conic domains are listed in Appendix A along with minimum sizes and possible parametric variations. While cones without parameter have a unique identifier, parametric cones are identified by an index into a lookup table written as `@INDEX:TABLE` in the CBF format. The index is numbered from zero, and the lookup tables are defined individually for each parametric cone type by a separate keyword. See Section C.3 for an example.

Keywords covered in this section:
`POWCONES` - *Define lookup table of power cone domains.*
`POW*CONES` - *Define lookup table of dual power cone domains.*

3.2 Problem structure

The problem structure defines the objective sense, whether it is minimization and maximization, using the keyword `OBJSENSE` (follow the hyperlink or see Appendix B). It also defines the index sets, $\mathcal{J}$, $\mathcal{J}^{PSD}$, $\mathcal{I}$ and $\mathcal{I}^{PSD}$, which are all numbered from zero, $\{0, 1, \ldots\}$, and empty until explicitly constructed.

- Scalar variables are constructed in vectors restricted to a conic domain, such as $(x_0, x_1) \in \mathbb{R}^2_+$, $(x_2, x_3, x_4) \in Q^3$, etc. In terms of the Cartesian product, this generalizes to $x \in \mathcal{K}_1^{n_1} \times \mathcal{K}_2^{n_2} \times \cdots \times \mathcal{K}_k^{n_k}$, which in the CBF format becomes

VAR
$n\ k$
$\mathcal{K}_1\ n_1$
$\mathcal{K}_2\ n_2$
$\ldots$
$\mathcal{K}_k\ n_k$

where n is the total number of scalar variables. The list of supported cones is explained in Section 3.1. Integrality of scalar variables can be specified afterwards, as in the minimal working example of Section 1.1, using the keyword INT (follow the hyperlink or see Appendix B).

- PSD variables are constructed one-by-one. That is, $X_j \in \texttt{PSD}^{n_j \times n_j}$ for $j \in \mathcal{J}^{PSD}$, construct matrix-valued variables of size $n_j \times n_j$ restricted to be symmetric positive semidefinite. In the CBF format, this list of constructions becomes

  PSDVAR
  N
  n_1
  n_2
  $\ldots$
  n_N

  where N is the total number of PSD variables.

- Scalar constraints are constructed in vectors restricted to a conic domain, such as $(g_0, g_1) \in \mathbb{R}^2_+$, $(g_2, g_3, g_4) \in Q^3$, etc. In terms of the Cartesian product, this generalizes to $g \in \mathcal{K}_1^{m_1} \times \mathcal{K}_2^{m_2} \times \cdots \times \mathcal{K}_k^{m_k}$, which in the CBF format becomes

  CON
  $m\ k$
  $\mathcal{K}_1\ m_1$
  $\mathcal{K}_2\ m_2$
  $\ldots$
  $\mathcal{K}_k\ m_k$

  where m is the total number of scalar constraints. The list of supported cones is explained in Section 3.1.

- PSD constraints are constructed one-by-one. That is, $G_i \in \texttt{PSD}^{m_i \times m_i}$ for $i \in \mathcal{I}^{PSD}$, construct matrix-valued affine expressions of size $m_i \times m_i$ restricted to be symmetric positive semidefinite. In the CBF format, this list of constructions becomes

  PSDCON
  M
  m_1
  m_2
  $\ldots$
  m_M

where M is the total number of PSD constraints.

With the objective sense, variables (with integer indications) and constraints, the definitions of the many affine expressions follow in problem data.

Keywords covered in this section:

`OBJSENSE` - *Define the objective sense.*
`VAR` - *Construct the scalar variables.*
`INT` - *Put integer requirements on a selected subset of scalar variables.*
`PSDVAR` - *Construct the PSD variables.*
`CON` - *Construct the scalar constraints.*
`PSDCON` - *Construct the PSD constraints.*

3.3 Problem data

The problem data defines the coefficients and constants of the affine expressions of the problem instance. These are considered zero until explicitly defined, implying that instances with no keywords from this information group are, in fact, valid. Duplicated or conflicting information is a failure to comply with the standard. Consequently, two coefficients written to the same position in a matrix (or to transposed positions in a symmetric matrix) is an error.

The affine expressions of the objective, g^{obj}, of the scalar constraints, g_i, and of the PSD constraints, G_i, are defined separately. The following notation uses the standard trace inner product for matrices, $\langle X, Y \rangle = \sum_{i,j} X_{ij} Y_{ij}$.

- The affine expression of the objective is defined as

$$g^{obj} = \sum_{j \in \mathcal{J}^{PSD}} \langle F_j^{obj}, X_j \rangle + \sum_{j \in \mathcal{J}} a_j^{obj} x_j + b^{obj},$$

  in terms of the symmetric matrices, F_j^{obj}, and scalars, a_j^{obj} and b^{obj}. These are specified on a sparse coordinate form using keywords `OBJFCOORD`, `OBJACOORD`, `OBJBCOORD` (follow the hyperlinks or see Appendix B).

- The affine expressions of the scalar constraints are defined, for $i \in \mathcal{I}$, as

$$g_i = \sum_{j \in \mathcal{J}^{PSD}} \langle F_{ij}, X_j \rangle + \sum_{j \in \mathcal{J}} a_{ij} x_j + b_i,$$

  in terms of the symmetric matrices, F_{ij}, and scalars, a_{ij} and b_i. These are specified on a sparse coordinate form using keywords `FCOORD`, `ACOORD`, and `BCOORD` (follow the hyperlinks or see Appendix B).

- The affine expressions of the PSD constraints are defined, for $i \in \mathcal{I}^{PSD}$, as

$$G_i = \sum_{j \in \mathcal{J}} x_j H_{ij} + D_i,$$

in terms of the symmetric matrices, H_{ij} and D_i. These are specified on a sparse coordinate form using keywords `HCOORD` and `DCOORD` (follow the hyperlinks or see Appendix B).

Keywords covered in this section:

`OBJFCOORD`	- *Define the affine expression of the objective.*
`OBJACOORD`	
`OBJBCOORD`	
`FCOORD`	- *Define the affine expressions of the scalar constraints.*
`ACOORD`	
`BCOORD`	
`HCOORD`	- *Define the affine expressions of the PSD constraints.*
`DCOORD`	

A List of cones

A distinction is made between *non-parametric cones* (Section A.1) that can be used directly, and *parametric cones* (Section A.2) that first have to be constructed for specific parameters.

A.1 Non-parametric cones

- **Free domain**
  CBF name: `F`.
  Dual cone: `L=`.
  Classification: Linear cone family.
  Definition:

$$\{x \in \mathbb{R}^n\}, \text{ for } n \geq 1.$$

- **Positive orthant**
  CBF name: `L+`.
  Dual cone: `L+`.
  Classification: Linear cone family.
  Definition:

$$\{x \in \mathbb{R}^n \mid x_j \geq 0 \text{ for } j = 1, \ldots, n\}, \text{ for } n \geq 1.$$

- **Negative orthant**
  CBF name: `L-`.
  Dual cone: `L-`.
  Classification: Linear cone family.
  Definition:

$$\{x \in \mathbb{R}^n \mid x_j \leq 0 \text{ for } j = 1, \ldots, n\}, \text{ for } n \geq 1.$$

- **Fixpoint zero**
  CBF name: `L=`.
  Dual cone: `F`.
  Classification: Linear cone family.
  Definition:

$$\{x \in \mathbb{R}^n \mid x_j = 0 \text{ for } j = 1, \ldots, n\}, \text{ for } n \geq 1.$$

- **One norm cone**
  CBF name: `ONENORM`.
  Dual cone: `INFNORM`.
  Classification: Linear cone family.
  Definition:

$$\left\{ \begin{pmatrix} t \\ x \end{pmatrix} \in \mathbb{R}^1_+ \times \mathbb{R}^{n-1} \mid t \geq \|x\|_1 \right\}, \text{ for } n \geq 1.$$

- **Infinity norm cone**
  CBF name: INFNORM.
  Dual cone: ONENORM.
  Classification: Linear cone family.
  Definition:

$$\left\{ \begin{pmatrix} t \\ x \end{pmatrix} \in \mathbb{R}^1_+ \times \mathbb{R}^{n-1} \mid t \geq \|x\|_\infty \right\}, \text{ for } n \geq 1.$$

- **Quadratic cone**
  CBF name: Q.
  Dual cone: Q.
  Classification: Second-order cone family.
  Definition:

$$\left\{ \begin{pmatrix} t \\ x \end{pmatrix} \in \mathbb{R}^1_+ \times \mathbb{R}^{n-1} \mid t \geq \sqrt{x^T x} \right\}, \text{ for } n \geq 1.$$

- **Rotated quadratic cone**
  CBF name: QR.
  Dual cone: QR.
  Classification: Second-order cone family.
  Definition:

$$\left\{ \begin{pmatrix} t \\ x \end{pmatrix} \in \mathbb{R}^2_+ \times \mathbb{R}^{n-2} \mid 2t_1 t_2 \geq x^T x \right\}, \text{ for } n \geq 2.$$

- **Semidefinite cone in symmetric vector form**
  CBF name: SVECPSD.
  Dual cone: SVECPSD.
  Classification: Semidefinite cone family.
  Definition:

$$\operatorname{svec}(\mathtt{PSD}^{n \times n}) \subseteq \mathbb{R}^{\frac{1}{2}n(n+1)}, \text{ for } n \geq 1,$$

  where

$$\operatorname{svec}(X) = [X_{11}, \sqrt{2}X_{21}, \ldots, \sqrt{2}X_{n1}, X_{22}, \sqrt{2}X_{32}, \ldots, \sqrt{2}X_{n2}, \ldots, X_{nn}], \quad (2)$$

  performs a column-wise extraction and off-diagonal scaling of the lower-triangular part of symmetric matrix-valued input to ensure inner product invariance; $\langle X, Y \rangle = \operatorname{svec}(X)^T \operatorname{svec}(Y)$.

- **Exponential cone**
  CBF name: EXP.
  Dual cone: EXP*.
  Classification: Exponential cone family.
  Definition:

$$\operatorname{cl}(S_1) = S_1 \cup S_2$$

where

$$S_1 = \left\{ \begin{pmatrix} t \\ s \\ r \end{pmatrix} \in \mathbb{R}^3 \mid s > 0,\ t \geq s \exp\left(\tfrac{r}{s}\right) \right\},$$
$$S_2 = \left\{ \begin{pmatrix} t \\ s \\ r \end{pmatrix} \in \mathbb{R}^3 \mid s = 0,\ t \geq 0,\ r \leq 0 \right\}.$$

- **Dual exponential cone**
  CBF name: EXP*.
  Dual cone: EXP.
  Classification: Exponential cone family.
  Definition:

$$\mathrm{cl}(S_1) = S_1 \cup S_2,$$

where

$$S_1 = \left\{ \begin{pmatrix} t \\ s \\ r \end{pmatrix} \in \mathbb{R}^3 \mid (-r) > 0,\ \mathbf{e}t \geq (-r) \exp\left(\tfrac{s}{r}\right) \right\},$$
$$S_2 = \left\{ \begin{pmatrix} t \\ s \\ r \end{pmatrix} \in \mathbb{R}^3 \mid \quad r = 0,\ \mathbf{e}t \geq 0,\ s \geq 0 \right\},$$

in terms of $\mathbf{e} = \exp(1)$.

- **Radial geometric mean cone**
  CBF name: GMEANABS.
  Dual cone: GMEANABS*.
  Classification: Power cone family.
  Definition:

$$\left\{ \begin{pmatrix} t \\ x \end{pmatrix} \in \mathbb{R}^k_+ \times \mathbb{R}^1 \mid \left(\prod_{j=1}^k t_j\right)^{1/k} \geq |x| \right\},\ \text{for } n = k + 1 \geq 2,$$

- **Dual radial geometric mean cone**
  CBF name: GMEANABS*.
  Dual cone: GMEANABS.
  Classification: Power cone family.
  Definition:

$$\left\{ \begin{pmatrix} t \\ x \end{pmatrix} \in \mathbb{R}^k_+ \times \mathbb{R}^1 \mid \left(\prod_{j=1}^k k t_j\right)^{1/k} \geq |x| \right\},\ \text{for } n = k + 1 \geq 2,$$

- **Geometric mean hypograph cone**

  CBF name: GMEAN.
  Dual cone: GMEAN*.
  Classification: Power cone family.
  Definition:

$$\left\{ \begin{pmatrix} t \\ x \end{pmatrix} \in \mathbb{R}^k_+ \times \mathbb{R}^1 \mid \left(\prod_{j=1}^k t_j\right)^{1/k} \geq x \right\}, \text{ for } n = k + 1 \geq 2,$$

- **Dual geometric mean hypograph cone**

  CBF name: GMEAN*.
  Dual cone: GMEAN.
  Classification: Power cone family.
  Definition:

$$\left\{ \begin{pmatrix} t \\ x \end{pmatrix} \in \mathbb{R}^k_+ \times \mathbb{R}^1 \mid \left(\prod_{j=1}^k k t_j\right)^{1/k} \geq -x \geq 0 \right\}, \text{ for } n = k + 1 \geq 2,$$

A.2 Parametric cones

- **Radial power cone**

  CBF name: POW.
  Dual cone: POW*.
  Classification: Power cone family.
  Definition: For any positive parameter $\alpha \in \mathbb{R}^k_{++}$,

  $$\left\{ \begin{pmatrix} t \\ x \end{pmatrix} \in \mathbb{R}^k_+ \times \mathbb{R}^{n-k} \mid \left(\prod_{j=1}^k t_j^{\alpha_j} \right)^{1/\sigma} \geq \|x\|_2 \right\}, \text{ for } n \geq k \geq 1,$$

  where $\sigma = \sum_{j=1}^k \alpha_j$.

- **Dual radial power cone**

  CBF name: POW*.
  Dual cone: POW.
  Classification: Power cone family.
  Definition: For any positive parameter $\alpha \in \mathbb{R}^k_{++}$,

  $$\left\{ \begin{pmatrix} t \\ x \end{pmatrix} \in \mathbb{R}^k_+ \times \mathbb{R}^{n-k} \mid \left(\prod_{j=1}^k \left((\alpha_j^{-1}\sigma) t_j \right)^{\alpha_j} \right)^{1/\sigma} \geq \|x\|_2 \right\}, \text{ for } n \geq k \geq 1,$$

  where $\sigma = \sum_{j=1}^k \alpha_j$.

- **Hypograph power cone**

  CBF name: POWH.
  Dual cone: POWH*.
  Classification: Power cone family.
  Definition: For any positive parameter $\alpha \in \mathbb{R}^k_{++}$,

  $$\left\{ \begin{pmatrix} t \\ x \end{pmatrix} \in \mathbb{R}^k_+ \times \mathbb{R}^1 \mid \left(\prod_{j=1}^k t_j^{\alpha_j} \right)^{1/\sigma} \geq x \right\}, \text{ for } n = k+1 \geq 2,$$

  where $\sigma = \sum_{j=1}^k \alpha_j$.

- **Dual hypograph power cone**

  CBF name: POWH*.
  Dual cone: POWH.
  Classification: Power cone family.
  Definition: For any positive parameter $\alpha \in \mathbb{R}^k_{++}$,

  $$\left\{ \begin{pmatrix} t \\ x \end{pmatrix} \in \mathbb{R}^k_+ \times \mathbb{R}^1 \mid \left(\prod_{j=1}^k \left((\alpha_j^{-1}\sigma) t_j \right)^{\alpha_j} \right)^{1/\sigma} \geq -x \geq 0 \right\}, \text{ for } n = k+1 \geq 2,$$

  where $\sigma = \sum_{j=1}^k \alpha_j$.

B List of keywords

All keywords are case sensitive and may not appear more than once in any instance specification. In summary, by information group, they are given as:

- **File format**

  `VER`

- **Parametric cone specification**

  `POWCONES`, `POW*CONES`

- **Problem structure**

  `OBJSENSE`, `PSDVAR`, `VAR`, `INT`, `PSDCON`, `CON`

- **Problem data**

  `OBJFCOORD`, `OBJACOORD`, `OBJBCOORD`
  `FCOORD`, `ACOORD`, `BCOORD`
  `HCOORD`, `DCOORD`
  `CHANGE`

The information groups must be ordered as specified in Section 2.2. Keywords, and their ordering within a given information group, are all optional unless explicitly stated as a remark to the respective keyword.

VER
The version of the Conic Benchmark Format used to write the file.

HEADER	None.
BODY	One line formatted as: `INT` This is the version number.

Remarks:
Must appear exactly once in a file, as the first keyword.

POWCONES

Define a lookup table of power cone domains.

HEADER: One line formatted as:
`INT INT`

This is the number of cones to be specified, and the combined length of their dense parameter vectors.

BODY: A list of chunks each specifying the dense parameter vector of a power cone.

- CHUNKHEADER: One line formatted as:
  `INT`

  This is the parameter vector length.

- CHUNKBODY: A list of lines formatted as
  `REAL`

  This is the parameter vector values. The number of lines should match the number stated in the chunk header.

The specified cone at index k (counted from 0) is registered under the CBF name `@k:POW`. The first and second number stated in the header should match the number of chunks and the sum of chunk header values, respectively.

POW*CONES

Define a lookup table of dual power cone domains.

HEADER: One line formatted as:
`INT INT`

This is the number of cones to be specified, and the combined length of their dense parameter vectors.

BODY: A list of chunks each specifying the dense parameter vector of a dual power cone.

- CHUNKHEADER: One line formatted as:
  `INT`

  This is the parameter vector length.

- CHUNKBODY: A list of lines formatted as
  `REAL`

  This is the parameter vector values. The number of lines should match the number stated in the chunk header.

The specified cone at index k (counted from 0) is registered under the CBF name `@k:POW*`. The first and second number stated in the header should match the number of chunks and the sum of chunk header values, respectively.

OBJSENSE

Define the objective sense.

HEADER None.

BODY One line formatted as:

```
STR
```

Having `MIN` indicates minimize, and `MAX` indicates maximize. Capital letters are required.

Remarks:
Must appear exactly once in a file.

PSDVAR

Construct the PSD variables.

HEADER One line formatted as:

```
INT
```

This is the number of PSD variables in the problem.

BODY A list of lines formatted as

```
INT
```

This indicates the number of rows (equal to the number of columns) in the matrix-valued PSD variable. The number of lines should match the number stated in the header.

VAR

Construct the scalar variables.

HEADER One line formatted as:

```
INT INT
```

This is the number of scalar variables, followed by the number of conic domains they are restricted to.

BODY A list of lines formatted as

```
STR INT
```

This indicates the cone name (see Appendix A), and the number of scalar variables restricted to this cone. These numbers should accumulate to the number of scalar variables stated first in the header. The number of lines should match the second number stated in the header.

INT

Put integer requirements on a selected subset of scalar variables.

HEADER — One line formatted as:

```
INT
```

This is the number of integer scalar variables in the problem.

BODY — A list of lines formatted as

```
INT
```

This indicates the scalar variable index $j \in \mathcal{J}$. The number of lines should match the number stated in the header.

Remarks:
Can only be used after these keywords: `VAR`

PSDCON

Construct the PSD constraints.

HEADER — One line formatted as:

```
INT
```

This is the number of PSD constraints in the problem.

BODY — A list of lines formatted as

```
INT
```

This indicates the number of rows (equal to the number of columns) in the matrix-valued affine expression of the PSD constraint. The number of lines should match the number stated in the header.

Remarks:
Can only be used after these keywords: `PSDVAR`, `VAR`

CON

Construct the scalar constraints.

HEADER — One line formatted as:

```
INT INT
```

This is the number of scalar constraints, followed by the number of conic domains they restrict to.

BODY — A list of lines formatted as

```
STR INT
```

This indicates the cone name (see Appendix A), and the number of affine expressions restricted to this cone. These numbers should accumulate to the number of scalar constraints stated first in the header. The number of lines should match the second number stated in the header.

Remarks:
Can only be used after these keywords: `PSDVAR`, `VAR`

OBJFCOORD

Input sparse coordinates (quadruplets) to define the symmetric matrices, F_j^{obj}, as used in the objective.

HEADER One line formatted as:

`INT`

This is the number of coordinates to be specified.

BODY A list of lines formatted as

`INT INT INT REAL`

This indicates the PSD variable index $j \in \mathcal{J}^{PSD}$, the row index, the column index and the coefficient value. The number of lines should match the number stated in the header.

OBJACOORD

Input sparse coordinates (pairs) to define the scalars, a_j^{obj}, as used in the objective.

HEADER One line formatted as:

`INT`

This is the number of coordinates to be specified.

BODY A list of lines formatted as

`INT REAL`

This indicates the scalar variable index $j \in \mathcal{J}$ and the coefficient value. The number of lines should match the number stated in the header.

OBJBCOORD

Input the scalar, b^{obj}, as used in the objective.

HEADER None.

BODY One line formatted as:

`REAL`

This indicates the coefficient value.

FCOORD

Input sparse coordinates (quintuplets) to define the symmetric matrices, F_{ij}, as used in the scalar constraints.

HEADER One line formatted as:

```
INT
```

This is the number of coordinates to be specified.

BODY A list of lines formatted as

```
INT INT INT INT REAL
```

This indicates the scalar constraint index $i \in \mathcal{I}$, the PSD variable index $j \in \mathcal{J}^{PSD}$, the row index, the column index and the coefficient value. The number of lines should match the number stated in the header.

ACOORD

Input sparse coordinates (triplets) to define the scalars, a_{ij}, as used in the scalar constraints.

HEADER One line formatted as:

```
INT
```

This is the number of coordinates to be specified.

BODY A list of lines formatted as

```
INT INT REAL
```

This indicates the scalar constraint index $i \in \mathcal{I}$, the scalar variable index $j \in \mathcal{J}$ and the coefficient value. The number of lines should match the number stated in the header.

BCOORD

Input sparse coordinates (pairs) to define the scalars, b_i, as used in the scalar constraints.

HEADER One line formatted as:

```
INT
```

This is the number of coordinates to be specified.

BODY A list of lines formatted as

```
INT REAL
```

This indicates the scalar constraint index $i \in \mathcal{I}$ and the coefficient value. The number of lines should match the number stated in the header.

HCOORD

Input sparse coordinates (quintuplets) to define the symmetric matrices, H_{ij}, as used in the PSD constraints.

HEADER One line formatted as:
`INT`
This is the number of coordinates to be specified.

BODY A list of lines formatted as
`INT INT INT INT REAL`
This indicates the PSD constraint index $i \in \mathcal{I}^{PSD}$, the scalar variable index $j \in \mathcal{J}$, the row index, the column index and the coefficient value. The number of lines should match the number stated in the header.

DCOORD

Input sparse coordinates (quadruplets) to define the symmetric matrices, D_i, as used in the PSD constraints.

HEADER One line formatted as:
`INT`
This is the number of coordinates to be specified.

BODY A list of lines formatted as
`INT INT INT REAL`
This indicates the PSD constraint index $i \in \mathcal{I}^{PSD}$, the row index, the column index and the coefficient value. The number of lines should match the number stated in the header.

CHANGE

Start of a new instance specification based on changes to the previous.

HEADER None.

BODY None.

Remarks:
Can be interpreted as the end of file when the hotstart-sequence is unsupported or undesired.

C Examples

C.1 Linear, second-order and semidefinite cones

The conic optimization problem (3), has a semidefinite cone, a quadratic cone over unordered subindices, and two equality constraints.

$$
\begin{array}{lll}
\text{minimize} & \left\langle \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}, X_0 \right\rangle + x_1 & \\
\text{subject to} & \left\langle \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, X_0 \right\rangle + x_1 & = \quad 1.0\,, \\
& \left\langle \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, X_0 \right\rangle + x_0 + x_2 & = \quad 0.5\,, \\
& x_1 \geq \sqrt{x_0^2 + x_2^2}\,, & \\
& X_0 \in \mathtt{PSD}^{3\times 3}\,. &
\end{array}
\tag{3}
$$

The equality constraints are easily rewritten to the conic form, $(g_0, g_1) \in \{0\}^2$, by moving constants such that the right-hand-side becomes zero. The quadratic cone does not fit under the `VAR` keyword in this variable permutation, however, as opposed to the minimal working example (1). Instead, it takes a scalar constraint $(g_2, g_3, g_4) = (x_1, x_0, x_2) \in Q^3$, with scalar variables constructed as $(x_0, x_1, x_2) \in \mathbb{R}^3$. Its formulation in the CBF format is shown below.

```
# File written using this version of the Conic Benchmark Format:
#     | Version 4.
VER
4

# The sense of the objective is:
#     | Minimize.
OBJSENSE
MIN

# One PSD variable of this size:
#     | Three times three.
PSDVAR
1
3

# Three scalar variables in this one conic domain:
#     | Three are free.
VAR
3 1
F 3
```

```
# Five scalar constraints with affine expressions in two conic domains:
#     | Two are fixed to zero.
#     | Three are in conic quadratic domain.
CON
5 2
L= 2
Q 3

# Five coordinates in F^{obj}_j coefficients:
#     | F^{obj}[0][0,0] = 2.0
#     | F^{obj}[0][1,0] = 1.0
#     | and more...
OBJFCOORD
5
0 0 0 2.0
0 1 0 1.0
0 1 1 2.0
0 2 1 1.0
0 2 2 2.0

# One coordinate in a^{obj}_j coefficients:
#     | a^{obj}[1] = 1.0
OBJACOORD
1
1 1.0

# Nine coordinates in F_ij coefficients:
#     | F[0,0][0,0] = 1.0
#     | F[0,0][1,1] = 1.0
#     | and more...
FCOORD
9
0 0 0 0 1.0
0 0 1 1 1.0
0 0 2 2 1.0
1 0 0 0 1.0
1 0 1 0 1.0
1 0 2 0 1.0
1 0 1 1 1.0
1 0 2 1 1.0
1 0 2 2 1.0

# Six coordinates in a_ij coefficients:
#     | a[0,1] = 1.0
#     | a[1,0] = 1.0
#     | and more...
ACOORD
6
0 1 1.0
1 0 1.0
1 2 1.0
2 1 1.0
3 0 1.0
4 2 1.0
```

```
# Two coordinates in b_i coefficients:
#     | b[0] = -1.0
#     | b[1] = -0.5
BCOORD
2
0 -1.0
1 -0.5
```

C.2 The exponential cone

The conic optimization problem (4) has one equality constraint, one quadratic cone constraint and one exponential cone constraint.

$$\begin{array}{lllll} \text{minimize} & x_0 - x_3 & & & \\ \text{subject to} & x_0 + 2x_1 - x_2 & \in & \{0\}, & \\ & \begin{pmatrix} 5.0 \\ x_0 \\ x_1 \end{pmatrix} & \in & Q^3, & \\ & \begin{pmatrix} x_2 \\ 1.0 \\ x_3 \end{pmatrix} & \in & \text{EXP}. & \end{array} \tag{4}$$

The nonlinear conic constraints enforce $\sqrt{x_0^2 + x_1^2} \leq 5$ and $x_3 \leq \log(x_2)$. Its formulation in the CBF format is shown below.

```
# File written using this version of the Conic Benchmark Format:
#     | Version 4.
VER
4

# The sense of the objective is:
#     | Minimize.
OBJSENSE
MIN

# Four scalar variables in this one conic domain:
#     | Four are free.
VAR
4 1
F 4

# Seven scalar constraints with affine expressions in three conic domains:
#     | One is fixed to zero.
#     | Three are in conic quadratic domain.
#     | Three are in exponential cone domain.
CON
7 3
L= 1
Q 3
EXP 3
```

```
# Two coordinates in a^{obj}_j coefficients:
#     | a^{obj}[0] = 1.0
#     | a^{obj}[3] = -1.0
OBJACOORD
2
0 1.0
3 -1.0

# Seven coordinates in a_ij coefficients:
#     | a[0,0] = 1.0
#     | a[0,1] = 2.0
#     | and more...
ACOORD
7
0 0 1.0
0 1 2.0
0 2 -1.0
2 0 1.0
3 1 1.0
4 2 1.0
6 3 1.0

# Two coordinates in b_i coefficients:
#     | b[1] = 5.0
#     | b[5] = 1.0
BCOORD
2
1 5.0
5 1.0
```

C.3 Using parametric cones

The problem (5) has three variables in a radial power cone with parameter $\alpha_1 = (1, 1)$, and two radial power cone constraints each with parameter $\alpha_0 = (8, 1)$.

$$
\begin{array}{lll}
\text{maximize} & x_2 & \\
\text{subject to} & \begin{pmatrix} 1.0 \\ x_0 \\ x_0 + x_1 \end{pmatrix} & \in \quad \mathrm{POW}_{\alpha_0}, \\
& \begin{pmatrix} 1.0 \\ x_1 \\ x_0 + x_1 \end{pmatrix} & \in \quad \mathrm{POW}_{\alpha_0}, \\
& x \in \mathrm{POW}_{\alpha_1}. &
\end{array}
\tag{5}
$$

The nonlinear constraints enforces $x_2 \leq x_0 x_1$ and $x_0 + x_1 \leq \min(x_0^{1/9}, x_1^{1/9})$. Its formulation in the CBF format is shown below.

```
# File written using this version of the Conic Benchmark Format:
#     | Version 4.
VER
4

# Two power cone domains defined in a total of four parameters:
#     | @0:POW (specification 0) has two parameters:
#         | alpha[0] = 8.0.
#         | alpha[1] = 1.0.
#     | @1:POW (specification 1) has two parameters:
#         | alpha[0] = 1.0.
#         | alpha[1] = 1.0.
POWCONES
2 4
2
8.0
1.0
2
1.0
1.0

# The sense of the objective is:
#     | Maximize.
OBJSENSE
MAX

# Three scalar variable in this one conic domain:
#     | Three are in power cone domain (specification 1).
VAR
3 1
@1:POW 3

# Six scalar constraints with affine expressions in two conic domains:
#     | Three are in power cone domain (specification 0).
#     | Three are in power cone domain (specification 0).
CON
6 2
@0:POW 3
@0:POW 3

# One coordinate in a^{obj}_j coefficients:
#     | a^{obj}[2] = 1.0
OBJACOORD
1
2 1.0

# Six coordinates in a_ij coefficients:
#     | a[1,0] = 1.0,
#     | a[2,0] = 1.0, and more...
ACOORD
6
1 0 1.0
2 0 1.0
2 1 1.0
4 1 1.0
5 0 1.0
5 1 1.0
```

```
# Two coordinates in b_i coefficients:
#     | b[0] = 1.0
#     | b[3] = 1.0
BCOORD
2
0 1.0
3 1.0
```

C.4 Mixing primal and dual standard form

The standard forms in semidefinite optimization are usually based either on semidefinite variables or linear matrix inequalities. In the CBF format, both forms are supported and can even be mixed as shown in (6).

$$
\begin{array}{lllr}
\text{minimize} & \left\langle \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X_0 \right\rangle + x_0 + x_1 + 1 & & \\
\text{subject to} & \left\langle \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, X_0 \right\rangle - x_0 - x_1 & \geq \quad 0.0\,, & \\
& \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix} x_0 + \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix} x_1 - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \in \quad \mathtt{PSD}^{2\times 2}\,, & \\
& X_0 \in \mathtt{PSD}^{2\times 2}\,. & &
\end{array}
\tag{6}
$$

Its formulation in the CBF format is shown below.

```
# File written using this version of the Conic Benchmark Format:
#     | Version 4.
VER
4

# The sense of the objective is:
#     | Minimize.
OBJSENSE
MIN

# One PSD variable of this size:
#     | Two times two.
PSDVAR
1
2

# Two scalar variables in this one conic domain:
#     | Two are free.
VAR
2 1
F 2
```

```
# One PSD constraint of this size:
#     | Two times two.
PSDCON
1
2

# One scalar constraint with its affine expression in this one conic domain:
#     | One is greater than or equal to zero.
CON
1 1
L+ 1

# Two coordinates in F^{obj}_j coefficients:
#     | F^{obj}[0][0,0] = 1.0
#     | F^{obj}[0][1,1] = 1.0
OBJFCOORD
2
0 0 0 1.0
0 1 1 1.0

# Two coordinates in a^{obj}_j coefficients:
#     | a^{obj}[0] = 1.0
#     | a^{obj}[1] = 1.0
OBJACOORD
2
0 1.0
1 1.0

# One coordinate in b^{obj} coefficient:
#     | b^{obj} = 1.0
OBJBCOORD
1.0

# One coordinate in F_ij coefficients:
#     | F[0,0][1,0] = 1.0
FCOORD
1
0 0 1 0 1.0

# Two coordinates in a_ij coefficients:
#     | a[0,0] = -1.0
#     | a[0,1] = -1.0
ACOORD
2
0 0 -1.0
0 1 -1.0

# Four coordinates in H_ij coefficients:
#     | H[0,0][1,0] = 1.0,
#     | H[0,0][1,1] = 3.0, and more...
HCOORD
4
0 0 1 0 1.0
0 0 1 1 3.0
0 1 0 0 3.0
0 1 1 0 1.0
```

```
# Two coordinates in D_i coefficients:
#     | D[0][0,0] = -1.0
#     | D[0][1,1] = -1.0
DCOORD
2
0 0 0 -1.0
0 1 1 -1.0
```

C.5 Using vectorized matrix cones

Semidefinite variables and linear matrix inequalities can both be reformulated in symmetric vector form which allows for semidefinite optimization without matrix-valued notation. Revisiting example (6) from Section C.4 and letting $x_{2:4} = \mathrm{svec}(X_0)$ in terms of (2), we find this equivalent formulation.

$$
\begin{array}{lll}
\text{minimize} & (x_2 + x_4) + x_0 + x_1 + 1 & \\
\text{subject to} & \sqrt{2}x_3 - x_0 - x_1 \quad \geq \quad 0.0\,, & \\
& \begin{pmatrix} 0 \\ \sqrt{2} \\ 3 \end{pmatrix} x_0 + \begin{pmatrix} 3 \\ \sqrt{2} \\ 0 \end{pmatrix} x_1 - \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \in \mathtt{SVECPSD}^3\,, & (7) \\
& x_{2:4} \in \mathtt{SVECPSD}^3\,. &
\end{array}
$$

Its formulation in the CBF format is shown below.

```
# File written using this version of the Conic Benchmark Format:
#     | Version 4.
VER
4

# The sense of the objective is:
#     | Minimize.
OBJSENSE
MIN

# Five scalar variables in two conic domains:
#     | Two are free.
#     | Three are PSD in symmetric vector form.
VAR
5 2
F 2
SVECPSD 3

# Four scalar constraints with affine expressions in two conic domains:
#     | One is greater than or equal to zero.
#     | Three are PSD in symmetric vector form.
CON
4 2
L+ 1
SVECPSD 3
```

```
# Four coordinates in a^{obj}_j coefficients:
#     | a^{obj}[0] = 1.0
#     | a^{obj}[1] = 1.0
#     | and more...
OBJACOORD
4
0 1.0
1 1.0
2 1.0
4 1.0

# One coordinate in b^{obj} coefficient:
#     | b^{obj} = 1.0
OBJBCOORD
1.0

# Seven coordinates in a_ij coefficients:
#     | a[0,0] = -1.0
#     | a[0,1] = -1.0
#     | and more...
ACOORD
7
0 0 -1.0
0 1 -1.0
0 3 1.4142135623730950488016887242l
1 1 3.0
2 0 1.4142135623730950488016887242l
2 1 1.4142135623730950488016887242l
3 0 3.0

# Two coordinates in b_i coefficients:
#     | b[1] = -1.0
#     | b[3] = -1.0
BCOORD
2
1 -1.0
3 -1.0
```

C.6 Optimizing over a sequence of objectives

The linear optimization problem (8), is defined for a sequence of objectives such that hotstarting from one to the next might be advantages.

$$
\begin{array}{lllll}
\text{maximize}_k & g_k^{obj} & & \\
\text{subject to} & 50\,x_0 + 31x_1 & \leq & 250\,, \\
 & 3\,x_0 - 2x_1 & \geq & -4\,, \\
 & x \in \mathbb{R}_+^2, & &
\end{array}
\tag{8}
$$

given,

$g_0^{obj} = x_0 + 0.64x_1.$

$g_1^{obj} = 1.11x_0 + 0.76x_1.$

$$g_2^{obj} = 1.11x_0 + 0.85x_1.$$

Its formulation in the CBF format is shown below.

```
# File written using this version of the Conic Benchmark Format:
#     | Version 4.
VER
4

# The sense of the objective is:
#     | Maximize.
OBJSENSE
MAX

# Two scalar variables in this one conic domain:
#     | Two are nonnegative.
VAR
2 1
L+ 2

# Two scalar constraints with affine expressions in these two conic domains:
#     | One is in the nonpositive domain.
#     | One is in the nonnegative domain.
CON
2 2
L- 1
L+ 1

# Two coordinates in a^{obj}_j coefficients:
#     | a^{obj}[0] = 1.0
#     | a^{obj}[1] = 0.64
OBJACOORD
2
0 1.0
1 0.64

# Four coordinates in a_ij coefficients:
#     | a[0,0] = 50.0
#     | a[1,0] = 3.0
#     | and more...
ACOORD
4
0 0 50.0
1 0 3.0
0 1 31.0
1 1 -2.0

# Two coordinates in b_i coefficients:
#     | b[0] = -250.0
#     | b[1] = 4.0
BCOORD
2
0 -250.0
1 4.0
```

```
# New problem instance defined in terms of changes.
CHANGE

# Two coordinate changes in a^{obj}_j coefficients. Now it is:
#     | a^{obj}[0] = 1.11
#     | a^{obj}[1] = 0.76
OBJACOORD
2
0 1.11
1 0.76

# New problem instance defined in terms of changes.
CHANGE

# One coordinate change in a^{obj}_j coefficients. Now it is:
#     | a^{obj}[0] = 1.11
#     | a^{obj}[1] = 0.85
OBJACOORD
1
1 0.85
```