User's guide of the MSSO-BlockIP package for multistage stochastic optimization *

Jordi Castro[1] Laureano F. Escudero[2] Juan F. Monge[3]

[1] Dept. of Statistics and Operations Research,
Universitat Politècnica de Catalunya, Barcelona, Catalonia
e-mail: jordi.castro@upc.edu

[2] Area of Statistics and Operations Research,
Universidad Rey Juan Carlos, Móstoles (Madrid), Spain
e-mail: laureano.escudero@urjc.es

[3] Center of Operations Research,
Universidad Miguel Hernández, Elche (Alicante), Spain
e-mail: monge@umh.es

Abstract

MSSO-BlockIP is a solver for continuous MultiStage Stochastic Optimization (MSSO) problems which relies on the specialized interior-point algorithm implemented in the BlockIP package. The class of MSSO problems dealt with by the solver includes those with either only strategical or both strategical and operational decisions, and with either linear or separable convex quadratic (i.e., with diagonal Hessian) objective functions. This short document presents the formulation and input format of MSSO problems solved by MSSO-BlockIP.

Key words: Multistage stochastic optimization, Interior-point methods, Strategic and tactical uncertainties, Large-scale optimization

1 Formulation of MSSO problems solved by MSSO-BlockIP

MSSO-BlockIP solves continuous (linear or separable convex quadratic) MultiStage Stochastic Optimization (MSSO) problems using an extension of the specialized interior-point method of [2, 1] for MSSO problems. MSSO-BlockIP nontrivially extends the approach initially introduced in [3] for two-stage stochastic optimization.

*This research has been supported by the projects RTI2018-097580-B-I00 (J. Castro), RTI2018-094269-B-I00 (L.F. Escudero) and MTM2016-79765-P (J.F. Monge)

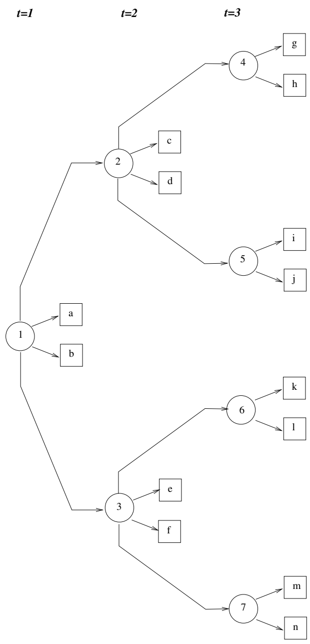

Figure 1: Strategic multistage scenario tree with operational two-stage scenario trees

We briefly introduce below the formulation of MSSO problems considered by the solver. A complete description of the formulation, algorithm and applications can be found in [4].

A MSSO problem with operational two-stage models, and its associated multistage scenario tree—as the one of Figure 1, which will be used for illustrative purposes—can be described by the following sets and parameters:

$\mathcal{T}$, set of stages ($\mathcal{T} = \{1, 2, 3\}$ in Figure 1).

$\mathcal{N}$, set of nodes in the scenario tree ($\mathcal{N} = \{1, \ldots, 7\}$ in Figure 1).

$\mathcal{N}_t$, set of nodes in stage t, where $\mathcal{N}_t \subset \mathcal{N}$, for $t \in \mathcal{T}$. By construction, $|\mathcal{N}_1| = \{1\}$. For instance, $\mathcal{N}_3 = \{4, 5, 6, 7\}$ in Figure 1.

Ω, set of scenarios. Each one is made by the nodes in the Hamiltonian path from the root node 1 to a node, say, ω in the last stage, through the stages in set $\mathcal{T}$; so, $\omega \in \mathcal{N}_{|\mathcal{T}|}$. For convenience, a scenario has traditionally been denoted by its last node in the path. For instance, $\Omega = \{4, 5, 6, 7\}$ in Figure 1.

$\Omega^n \subseteq \Omega$, scenarios containing node n in the path from root node 1 to ω. Note that $\Omega^1 = \Omega$.

$\mathcal{S}^n$, strategical successor nodes of node n, for $n \in \mathcal{N}$. Note that $\mathcal{S}^n = \emptyset$, for $n \in \mathcal{N}_{|\mathcal{T}|}$; and $\mathcal{S}^1 = \mathcal{N} \setminus \{1\}$.

$\mathcal{S}_1^n \subseteq \mathcal{S}^n$**,** immediate strategical successor nodes of node n, for $n \in \mathcal{N}$. For instance, $\mathcal{S}^3 = \mathcal{S}_1^3 = \{6, 7\}$ and $\mathcal{S}_1^1 = \{2, 3\}$ in Figure 1.

w^n**,** weight factor representing the likelihood that is associated with node n, for $n \in \mathcal{N}$. Note that $w^n = \sum_{\omega \in \Omega^n} w^\omega$, where w^ω gives the modeler-driven likelihood associated with scenario ω, such that $\sum_{\omega \in \Omega} w^\omega = 1$.

t^n**,** stage to which node n belongs to, so, $n \in \mathcal{N}_{t^n}$. For instance, $t^7 = 3$ in Figure 1.

σ^n**,** immediate ancestor node of node n, for $n \in \mathcal{N}$. Note: It is assumed that $\sigma^1 = \emptyset$. For instance, $\sigma^6 = 3$ in Figure 1.

$s(1)$**,** first strategical node in set $\mathcal{S}_1^n : t^n < |\mathcal{T}|$, for $n \in \mathcal{N}$. For instance, for node 1 of Figure 1, $s(1) = 2$.

$s(\ell)$**,** last strategical node in set $\mathcal{S}_1^n : t^n < |\mathcal{T}|$, for $n \in \mathcal{N}$ (where ℓ is the number of child nodes for every node). For instance, $\ell = 2$ in Figure 1; and for node 1, $s(2) = 3$.

$s(i)$**,** i-th strategical node in set $\mathcal{S}_1^n : t^n < |\mathcal{T}|$, for $n \in \mathcal{N}$, $i = 1, \ldots \ell$.

Π^n**,** set of operational scenarios for strategic node $n \in \mathcal{N}$. The number of operational scenarios $|\Pi^n|$ is the same for all $n \in \mathcal{N}$, and it will be named $|\Pi|$. In Figure 1 operational nodes are in squared boxes, and we have two per strategic node: for instance, $\Pi^1 = \{\text{a}, \text{b}\}$, $\Pi^7 = \{\text{m}, \text{n}\}$.

$\pi(i)$**,** i-th operational node in set Π^n, for $n \in \mathcal{N}$, $i = 1, \ldots, |\Pi^n|$. For instance, for node 1, $\pi(1) = \text{a}$ and $\pi(2) = \text{b}$.

w^π**,** weight or probability of operational scenario π, for $\pi \in \Pi^n$, such that $\sum_{\pi \in \Pi^n} w^\pi = 1$, for $t \in \mathcal{T}$.

Using the above definition of sets and parameters, the compact version of the strategic multistage operational two-stage meta model to consider can be formulated as

$$\min_{x^n, z^n, y_n^\pi} \quad \sum_{n \in \mathcal{N}} w^n \Big[a^n x^n + \frac{1}{2} x^{n\top} Q_x^n x^n + b^n z^n + \frac{1}{2} z^{n\top} Q_z^n z^n \quad + \sum_{\pi \in \Pi^n} \big(w^\pi c_n^\pi y_n^\pi + \frac{1}{2} y_n^{\pi\top} Q_n^\pi y_n^\pi \big) \Big] \tag{1a}$$

$$\begin{aligned}
\text{s.to} \quad & T^n x^{\sigma^n} + W^n x^n + M^n z^n = h^n && \forall n \in \mathcal{N} && \text{(1b)} \\
& T_n^\pi x^n + W_n^\pi y_n^\pi = h_n^\pi && \forall \pi \in \Pi^n,\ n \in \mathcal{N} && \text{(1c)} \\
& 0 \le x^n \le u_x^n,\ 0 \le z^n \le u_z^n && \forall n \in \mathcal{N} && \text{(1d)} \\
& 0 \le y_n^\pi \le u_{y,n}^\pi && \forall \pi \in \Pi^n,\ n \in \mathcal{N}, && \text{(1e)}
\end{aligned}$$

where a^n, b^n, c_n^π and Q_x^n, Q_z^n, Q_n^π are the vectors and diagonal matrices of the linear and quadratic terms of the objective function for the variables x^n, z^n and y_n^π, respectively; T^n and W^n are the constraint matrices of the state strategic variables x^{σ^n} in the first stage and

x^n in the second stage strategic node n, respectively; M^n is the constraint matrix of the local strategic variables z^n; T^π_n and W^π_n are the constraint matrices of the state strategic variables x^n in the first stage and the operational variables y^π_n in the second stage of the two-stage operational scenario π in the related embedded operational two-stage submodels, respectively; h^n and h^π are the *rhs* of the two-stage strategic and operational constraints, resp., u^n_x, u^n_z and $u^\pi_{y,n}$ are the upper bounds of the variables in the vectors x^n, z^n and y^π_n, resp.

MSSO-BlockIP, see [4] for details, uses a variable splitting formulation with the following copies of the variables:

x^s_n, copy of x^n in strategic node s, where n is the strategic node that roots the strategic two-stage tree, and s is a second stage node, for $s \in \mathcal{S}^n_1$, $n \in \mathcal{N} : t^n < T$.

x^π_n, copy of x^n in operational node π, where n is the strategic node that roots the operational two-stage tree and π is a second stage node, for $\pi \in \Pi^n$.

Thus, the splitting variable formulation of meta model (1), needed by MSSO-BlockIP is:

$$\min_{x^n,z^n,y^\pi_n} \quad \sum_{n\in\mathcal{N}} w^n \big[a^n x^n + \frac{1}{2} x^{n\top} Q^n_x x^n + \quad b^n z^n + \frac{1}{2} z^{n\top} Q^n_z z^n + \sum_{\pi\in\Pi^n} \big(w^\pi c^\pi_n y^\pi_n + \frac{1}{2} y^{\pi\top}_n Q^\pi_n y^\pi_n \big) \big] \tag{2a}$$

$$\begin{aligned}
\text{s.to} \quad & x^n - x^{s(1)}_n = 0 && \forall n \in \mathcal{N}_t : t < T && \text{(2b)}\\
& x^{s(i)}_n - x^{s(i+1)}_n = 0 && \forall i = 1,\dots,\ell-1,\ n \in \mathcal{N} : t^n < T && \text{(2c)}\\
& T^n x^n_{\sigma^n} + W^n x^n + M^n z^n = h^n && \forall n \in \mathcal{N} && \text{(2d)}\\
& x^{s(\ell)}_n - x^{\pi(1)}_n = 0 && \forall n \in \mathcal{N} && \text{(2e)}\\
& x^{\pi(i)}_n - x^{\pi(i+1)}_n = 0 && \forall i = 1,\dots,|\Pi^n|-1,\ n \in \mathcal{N} && \text{(2f)}\\
& T^\pi_n x^\pi_n + W^\pi_n y^\pi_n = h^\pi && \forall \pi \in \Pi^n,\ n \in \mathcal{N} && \text{(2g)}\\
& 0 \le x^n \le u^n_x,\ 0 \le z^n \le u^n_z && \forall n \in \mathcal{N} && \text{(2h)}\\
& 0 \le x^{s(i)}_n \le u^n_x, && \forall i = 1,\dots,\ell,\ n \in \mathcal{N} && \text{(2i)}\\
& 0 \le x^{\pi(i)}_n \le u^n_x, && \forall i = 1,\dots,|\Pi^n|,\ n \in \mathcal{N} && \text{(2j)}\\
& 0 \le y^\pi_n \le u^\pi_{y,n} && \forall \pi \in \Pi^n,\ n \in \mathcal{N}. && \text{(2k)}
\end{aligned}$$

Since MSSO-BlockIP relies on the BlockIP solver, problem (2) must be recast in the standard primal block angular form of BlockIP:

$$\min_{\mathbf{x}^1,\dots,\mathbf{x}^k} \sum_{i=1}^{k} \Big((\mathbf{c}^i)^\top \mathbf{x}^i + (\mathbf{x}^i)^\top Q^i \mathbf{x}^i \Big)$$

$$\text{s. to} \quad \begin{bmatrix} N_1 & & & & \\ & N_2 & & & \\ & & \ddots & & \\ & & & N_k & \\ R_1 & R_2 & \dots & R_k & I \end{bmatrix} \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{x}^2 \\ \vdots \\ \mathbf{x}^k \\ \mathbf{x}^0 \end{bmatrix} = \begin{bmatrix} \mathbf{b}^1 \\ \mathbf{b}^2 \\ \vdots \\ \mathbf{b}^k \\ \mathbf{b}^0 \end{bmatrix} \tag{3}$$

$$0 \le \mathbf{x}^i \le \mathbf{u}^i \quad i = 0, \dots, k.$$

Matrices N_i and R_i, $i = 1, \dots, k$ are related to block and linking constraints, respectively, k being the number of blocks. Vector $\mathbf{x}^i$ contains the variables of block i, for $i = 1, \dots, k$. Vector $\mathbf{x}^0$ has the slacks of the linking constraints; if they are equalities we set $\mathbf{u}^0 = 0$ (in practice 0 is replaced by a very small tolerance). Both linear and convex quadratic separable costs, defined by vectors $\mathbf{c}^i$ and (diagonal and positive semidefinite) matrices Q^i are considered.

Model (2) can be rewritten to match the standard form (3) by considering an appropriate reordering of the variables and constraints. This reordering is also instrumental for the performance of the interior-point algorithm since it avoids fill-in for the linking constraints (see [4] for details). The order of variables considered is based on a breadth-first-search (BFS) of the scenario tree (that is, nodes are explored by stages). And for each node n in the BFS, the first variables considered are those involved in the (nested) two-stage tree made by node n and its successor nodes $s \in \mathcal{S}_1^n$ (including all copies x_n^s, local variables z^s, and state variables x^s), followed by the variables in the operational tree made by n and $\pi \in \Pi^n$ (that is, x_n^π and y_n^π). For instance, for the scenario tree of Figure 1 the constraints matrix in standard form (3) is shown in Figure 2 for only the strategic tree, and in Figure 3 for the strategic and operational tree. In these figures, vertical lines separate the k blocks of the problem; a double horizontal line separates the diagonal block constraints $N_i \mathbf{x}^i = \mathbf{b}^i$ from the linking constraints, and, within the linking constraints part, a horizontal line separates the constraints of different—nested—two-stage trees. Note that in general the number of blocks is $k = \dfrac{\ell^{|\mathcal{T}|} - 1}{\ell - 1}(|\Pi| + 1)$ (e.g., in Figure 2 we have $\ell = 2$, $|\mathcal{T}| = 3$ and $|\Pi| = 0$, so $k = 2^3 - 1 = 7$; while in Figure 3 $|\Pi| = 2$ and $k = 21$).

The MSSO problem must be provided to MSSO-BlockIP in the input format described in next Section 2.

2 Input format for MSSO problems

The user must provide information about the structure of the scenario tree (stages, number of strategic and operational nodes, etc), and all the data needed to build problem (3),

x^1	z^1	x_1^2	x^2	z^2	x_1^3	x^3	z^3	x_2^4	x^4	z^4	x_2^5	x^5	z^5	x_3^6	x^6	z^6	x_3^7	x^7	z^7
W^1	M^1																		
		T^2	W^2	M^2															
					T^3	W^3	M^3												
								T^4	W^4	M^4									
											T^5	W^5	M^5						
														T^6	W^6	M^6			
																	T^7	W^7	M^7
I		$-I$																	
		I			$-I$														
			I					$-I$											
								I			$-I$								
						I								$-I$					
														I			$-I$		

Figure 2: Constraints structure for scenario tree of Figure 1 considering only strategic nodes

that is, block matrices N_i, rhs vectors $\mathbf{b}^i$, linear and quadratic costs $\mathbf{c}^i$ and Q^i, and upper bounds $\mathbf{u}^i$ for $i = 1, \ldots, k$. The order of the blocks, and the order of the variables within the blocks, must match the one illustrated in Figures 2 and 3. Vectors and matrices related to linking constraints (that is, $\mathbf{b}^0$, $\mathbf{u}^0$ and R_i for $i = 1, \ldots, k$) do not need to be provided, since they are internally dealt with by MSSO-BlockIP from the information about the tree of scenarios. This information is provided through a text file with the following format:

```
#Lines starting with # are not read, and they can be used to comment the file.
#Number of stages |T|
```
$|\mathcal{T}|$
```
#Number of strategical successor nodes per node
```
ℓ
```
#Number of operational nodes; if 0, no operational tree considered
```
$|\Pi|$
```
#For each stage t= 1..|T|, number nvstr_t of strategical variables (state variables x
#and local variables z) in the nodes of stage t. The number of variables
#to be copied (the x) must be less or equal than nvstr
```
nvstr_1
nvstr_2
$\vdots$
$\mathrm{nvstr}_{|\mathcal{T}|}$
```
#For each stage t= 1..|T|, number nvsplit_t of state variables x to be splitted
#in nodes of stage t. Note that nvsplit_t <= nvstr_t
```
$\mathrm{nvsplit}_1$
$\mathrm{nvsplit}_2$
$\vdots$
$\mathrm{nvsplit}_{|\mathcal{T}|}$
```
#For each stage t= 1..|T|, number of local variables y in operational nodes of stage t
#If the number of operational nodes is 0, nothing to be read (no line needed)
```

x^1	z^1	x^2_1	x^2	z^2	x^3_1	x^3	z^3	x^a_1	y^a_1	x^b_1	y^b_1	x^4_2	x^4	z^4	x^5_2	x^5	z^5	x^c_2	y^c_2	x^d_2	y^d_2	x^6_3	x^6	z^6	x^7_3	x^7	z^7	x^e_3	y^e_3	x^f_3	y^f_3	x^g_4	y^g_4	x^h_4	y^h_4	x^i_5	y^i_5	x^j_5	y^j_5	x^k_6	y^k_6	x^l_6	y^l_6	x^m_7	y^m_7	x^n_7	y^n_7
W^1	M^1																																													
		T^2	W^2	M^2																																										
					T^3	W^3	M^3																																							
								T^{a}_1	W^{a}_1																																					
										T^{b}_1	W^{b}_1																																			
												T^4	W^4	M^4																																
															T^5	W^5	M^5																													
																		T^{c}_2	W^{c}_2																											
																				T^{d}_2	W^{d}_2																									
																						T^6	W^6	M^6																						
																									T^7	W^7	M^7																			
																												T^{e}_3	W^{e}_3																	
																														T^{f}_3	W^{f}_3															
																																T^{g}_4	W^{g}_4													
																																		T^{h}_4	W^{h}_4											
																																				T^{i}_5	W^{i}_5									
																																						T^{j}_5	W^{j}_5							
																																								T^{k}_6	W^{k}_6					
																																										T^{l}_6	W^{l}_6			
																																												T^{m}_7	W^{m}_7	
																																														T^{n}_7	W^{n}_7
I		$\begin{matrix}-I\\ I\end{matrix}$			$\begin{matrix}-I\\ I\end{matrix}$			$\begin{matrix}-I\\ I\end{matrix}$		$-I$																																				
			I									$\begin{matrix}-I\\ I\end{matrix}$			$\begin{matrix}-I\\ I\end{matrix}$			$\begin{matrix}-I\\ I\end{matrix}$		$-I$																										
						I																$\begin{matrix}-I\\ I\end{matrix}$			$\begin{matrix}-I\\ I\end{matrix}$			$\begin{matrix}-I\\ I\end{matrix}$		$-I$																
													I																			$\begin{matrix}-I\\ I\end{matrix}$		$-I$												
																I																				$\begin{matrix}-I\\ I\end{matrix}$		$-I$								
																							I																	$\begin{matrix}-I\\ I\end{matrix}$		$-I$				
																										I																		$\begin{matrix}-I\\ I\end{matrix}$		$-I$	

Figure 3: Constraints structure for scenario tree of Figure 1 considering strategic and operational nodes

nvoper_1
nvoper_2
$\vdots$
$\text{nvoper}_{|\mathcal{T}|}$
```
#Number of blocks k
```
k
```
#For each block i=1..k, matrix N_i. The structure of N_i depends of block i.
#For instance, N_1= [W^1 M^1], N_j= [T^s W^s M^s] if j is block of strategical node s,
#N_j= [T^p W^p] if j is block of operational node p
#N_1
```
Here goes N_1 (see below input format for matrices)
```
#N_2
```
Here goes N_2 (see below input format for matrices)
$\vdots$
```
#N_k
```
Here goes N_k (see below input format for matrices)
```
#Type of objective function:  0=linear 1=quadratic
```
0 or 1
```
#For each block i=1..k, linear and quadratic costs c^i and Q^i
#If objective is linear leave empty Q entries
#Ordered by blocks, and by variable within blocks
```
$\mathbf{c}_1^1 \quad Q_{11}^1$
$\vdots$
$\mathbf{c}_{n_1}^1 \quad Q_{n_1,n_1}^1$ (n_1 is the number of variables of block 1)
$\mathbf{c}_1^2 \quad Q_{11}^2$
$\vdots$
$\mathbf{c}_{n_2}^2 \quad Q_{n_2,n_2}^2$ (n_2 is the number of variables of block 2)
$\vdots$
$\mathbf{c}_1^k \quad Q_{11}^k$
$\vdots$
$\mathbf{c}_{n_k}^k \quad Q_{n_k,n_k}^k$ (n_k is the number of variables of block k)
```
#For each block i=1..k, upper bounds u^i
#Ordered by blocks, and by variable within blocks
```
$\mathbf{u}_1^1$
$\vdots$
$\mathbf{u}_{n_1}^1$ (n_1 is the number of variables of block 1)
$\mathbf{u}_1^2$
$\vdots$
$\mathbf{u}_{n_2}^2$ (n_2 is the number of variables of block 2)
$\vdots$
$\mathbf{u}_1^k$

$\vdots$

$\mathbf{u}^k_{n_k}$ (n_k is the number of variables of block k)

```
#For each block i=1..k, right-hand-sides b^i
#Ordered by blocks, and by variable within blocks
```

$\mathbf{b}^1_1$

$\vdots$

$\mathbf{b}^1_{m_1}$ (m_1 is the number of constraints of block 1)

$\mathbf{b}^2_1$

$\vdots$

$\mathbf{b}^2_{m_2}$ (m_2 is the number of constraints of block 2)

$\vdots$

$\mathbf{b}^k_1$

$\vdots$

$\mathbf{b}^k_{m_k}$ (m_k is the number of constraints of block k)

The format for matrices N^i, $i = 1, \ldots, k$ is as follows:

```
#Format for a general sparse matrix M.
#First line gives m,n,nnz: number of rows, columns and nonzeros in matrix.
#Next h=1,...,nnz lines provide i(h),j(h),a(h): row, column and value of h entry.
#One-indexed arrays considered, so  1<=i(h)<=m, 1<=j(h)<=n for h=1,...,nnz.
```

$m\ n\ nnz$

$i_1\ j_i\ a_1$

$\vdots$

$i_{nnz}\ j_{nnz}\ a_{nnz}$

3 Usage of MSSO-BlockIP

To solve a problem with MSSO-BlockIP we just run:

```
MSSO-BlockIP input_file  {options}
```

The parameter `input_file` contains the MSSO problem (in the format described in Section 2). The available options, most of them related to the specialized interior-point algorithm implemented in BlockIP, are (see [1, 2] for details):

```
-out {name}
  Output file for optimal variables (if not provided, solution is not reported)
-mps {name}
  If name is provided, an mps format of the problem is written to filename 'name'
-only_mps {name}
  If name is provided, the program stops after  writing mps format (problem
```

```
  is not solved)
-inf {value}
  Set infinity value to be considered
-ub_slacks_linking {double >0}
  Upper bound of slacks of linking constraints of splitted variables, must be close to 0. Defa
-m_pw_prec {value}
  Set number of terms used as preconditioner of the power series expansion
  of (D-C'*B^{-1}*B)^{-1}
-sigma {value}
  Set reduction of the centrality parameter at each IP iteration
-rho {value}
  Set reduction of the step-length for the primal and dual variables at each
  IP iteration
-optim_gap {value}
  Set optimality gap tolerance
-optim_pfeas {value}
  Set primal feasibility tolerance
-optim_dfeas {value}
  Set dual feasibility tolerance
-output_freq {value}
  Set output information lines will be printed each output_freq IP iterations
-output {value}
  Set type of output. Can take the values: 0= NONE, 1= SCREEN, 2= FILE, 3=BOTH
-maxiter {value}
  Set maximum number of IP iterations
-min_pcgtol {value}
  Set minimum tolerance for the conjugate gradient
-red_pcgtol {value}
  Set reduction of pcg_tol at each IP iteration
-init_pcgtol {value}
  Set initial tolerance for the conjugate gradient (by default PCG_TOL_LIN
  if linear problem, PCG_TOL_QUAD if quadratic or -convex- nonlinear)
-type_start_point {value}
  Set how starting point is computed. Can take the values:
  0= SIMPLE, 1= QUAD_EQCONS_PROB
-type_comp_dy {value}
  Set how dy direction is computed. Can take the values: 0= CHOL_PWRS_PCG,
  1= FULL_CHOL, 2= HYBRID_PCG, 3= CHOL_THETA0_PCG, 4= PWRS_THETA0_PCG,
  5=THETA0_PWRS_PCG, 6=AUT_PWRSTHETA0_PCG, 7=DYN_PWRSTHETA0_PCG
-type_direction {value}
  Set which direction is computed. Can take the values: 0= NEWTON,
  1= SECOND_ORDER, 2= AUTOMATIC
-deactivateLnk {value}
  Set flag on deactivation of linking
-type_reg {value}
  Set type of regularization performed. Can take the values: 0=NO_REG,
  1= QUAD_REG, 2= PROX_REG
-factor_reg {value}
  Set initial value for regularization term
```

```
-show_specrad {y/n}
  Whether estimation of spectral radius by Ritz values is shown
  at each IP iteration
-show_princ_angles {y/n}
  Whether average principal angles between L and N are shown when computed
-threshold_angle {value}
  Set threshold principal angle to switch between PWRS and THETA0 preconditioners
-threshold_specrad {value}
  Set threshold spectral radius to switch between PWRS and THETA0 preconditioners
-it_Theta0PWRS {value}
  Set number of iterations between checks to switch between PWRS and THETA0
-type_comp_angle {value}
  Set how principal angles are computed. Possible values are:
  0= LINEARCOMB, 1= AVERAGE
-gap_changeChol {value}
  Change to Cholesky if duality gap below this value
-zero_pivots {value}
  Max number of zero pivots allowed in Cholesky factorization
-show_zero_pivots {y/n}
  Show_zero_pivots at each IP iteration
-stop_if_PCG_fails {y/n}
  If PCG fails stop, do not switch to Cholesky, useful if instance is large
-freevars {0/1}
  How to deal with free variables: 0=SPLIT , 1=REGULARIZE
-threads n
  Set number of threads (by default 1; if n<=1, 1; otherwise
  min{n,max number of threads})
```

References

[1] J. Castro (2016). Interior-point solver for convex separable block-angular problems. *Optimization Methods and Software*, 31:88-109.

[2] J. Castro & J. Cuesta (2011). Quadratic regularizations in an interior-point method for primal block-angular problems. *Mathematical programming*, 130:415-445.

[3] J. Castro & P. de la Lama-Zubirán (2020). A new interior-point approach for large two-stage stochastic problems. *Optimization Methods and Software*, doi: 10.1080/10556788.2020.1841190

[4] J. Castro & L.F. Escudero & J.F. Monge (2021). On solving large multistage stochastic problems with a new specialized interior-point approach. Submitted.