Nonconvex Generalized Benders Decomposition for Stochastic Separable Mixed-Integer Nonlinear Programs - Supplementary Material

A Scenario Generation for Normal Distribution

The naive sampling rule used to generate scenarios for normal distribution in the case studies is introduced here. If the normal distribution has mean μ and standard deviation σ and the total number of scenarios to be sampled is s, then scenarios are only sampled in the range $[-3\sigma+\mu, 3\sigma+\mu]$ and the hth scenario is

$$r_h = -3\sigma+\mu+3\sigma/s+(h-1)6\sigma/s.$$

This is illustrated in Figure A.1. Also, the probability of each scenario is

$$P(r=r_h) = \begin{cases} \Phi^{-1}\left(-3\sigma+\mu+6\sigma/s\right), & \text{if } h=1, \\ \Phi^{-1}\left(-3\sigma+\mu+6\sigma h/s\right) - \Phi^{-1}\left(-3\sigma+\mu+6\sigma(h-1)/s\right), & \text{if } 1<h<s, \\ 1-\Phi^{-1}\left(-3\sigma+\mu+6\sigma(s-1)/s\right), & \text{if } h=s, \end{cases}$$

where Φ^{-1} denotes the inverse cumulative distribution function of the normal distribution.

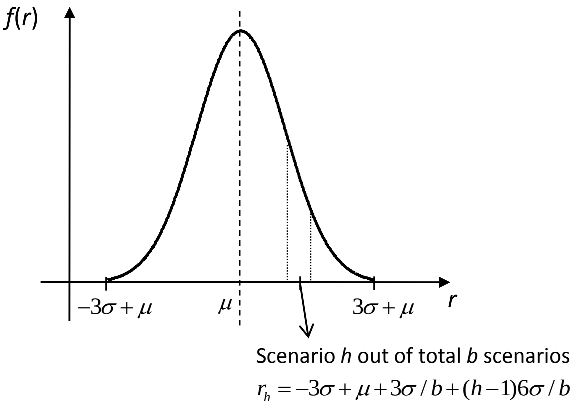

Figure A.1 Scenario generation for normal distribution.

B McCormick Relaxation Details for the Case Studies

The univariate intrinsic functions in the case studies include logarithmic functions, quadratic functions and cubic functions, and their convex and concave envelopes are given in Table B.1. The relaxation rule for products is given in Table B.2

Table B.1 Convex and concave envelopes of some univariate intrinsic functions

Univariate Intrinsic Function	Bounds on variables	Convex and Concave Envelope
$z = \ln(x)$	$x^{\mathrm{lo}} \leq x \leq x^{\mathrm{up}}$	$z \leq \ln(x)$ $z \geq \frac{\ln(x^{\mathrm{up}}) - \ln(x^{\mathrm{lo}})}{x^{\mathrm{up}} - x^{\mathrm{lo}}}(x - x^{\mathrm{up}}) + \ln(x^{\mathrm{up}})$
$z = x^2$	$x^{\mathrm{lo}} \leq x \leq x^{\mathrm{up}}$	$z \leq (x^{\mathrm{up}} + x^{\mathrm{lo}})x - x^{\mathrm{up}}x^{\mathrm{lo}}$ $z \geq x^2$
$z = x^3$	$0 \leq x^{\mathrm{lo}} \leq x \leq x^{\mathrm{up}}$	$z \leq \frac{(x^{\mathrm{up}})^3 - (x^{\mathrm{lo}})^3}{x^{\mathrm{up}} - x^{\mathrm{lo}}}(x - x^{\mathrm{up}}) + (x^{\mathrm{up}})^3$ $z \geq x^3$

Table B.2 McCormick relaxation rule for products

	Bounds on variables or functions	Convex and Concave Envelope
$v = z_1 z_2$	$z_1^{\mathrm{lo}} \leq z_1 \leq z_1^{\mathrm{up}}$ $z_2^{\mathrm{lo}} \leq z_2 \leq z_2^{\mathrm{up}}$	$v \geq z_1^{\mathrm{up}} z_2 + z_1 z_2^{\mathrm{up}} - z_1^{\mathrm{up}} z_2^{\mathrm{up}}$ $v \geq z_1^{\mathrm{lo}} z_2 + z_1 z_2^{\mathrm{lo}} - z_1^{\mathrm{lo}} z_2^{\mathrm{lo}}$ $v \leq z_1^{\mathrm{lo}} z_2 + z_1 z_2^{\mathrm{up}} - z_1^{\mathrm{lo}} z_2^{\mathrm{up}}$ $v \leq z_1^{\mathrm{up}} z_2 + z_1 z_2^{\mathrm{lo}} - z_1^{\mathrm{up}} z_2^{\mathrm{lo}}$

Table C.1 Nominal parameter values for the software reliability problem

Parameter	Index i	Index j	value	Parameter	Index i	Index j	value
R	1	1	0.90	E	1	1	3
R	1	2	0.80	E	1	2	1
R	1	3	0.85	E	1	3	2
R	2	1	0.95	E	2	1	3
R	2	2	0.80	E	2	2	2
R	2	3	0.85	E	2	3	1
R	3	1	0.98	E	3	1	3
R	3	2	0.94	E	3	2	2
				E_{tot}			10

C The Software Reliability Problem

According to its deterministic formulation in [44], the stochastic formulation for the problem can be written as follows:

$$\begin{aligned}
&\max_{y_{i,j},r_{i,h}} \sum_{h=1}^{s} w_h \left(\prod_{i=1}^{n} r_{i,h} \right) \\
&\text{s.t.} \quad -\ln(1-r_{i,h}) + \sum_{j=1}^{m_i} y_{i,j} \ln(1-R_{i,j,h}) = 0, \quad i=1,\dots,n,\ h=1,\dots,s \\
&\qquad 0 \le r_{i,h} \le 1 - \prod_{j=1}^{m_i}(1-R_{i,j,h}), \quad i=1,\dots,n,\ h=1,\dots,s \qquad \text{(SRP-S)} \\
&\qquad \sum_{j=1}^{m_i} y_{i,j} \ge 1, \quad i=1,\dots,n, \\
&\qquad \sum_{i=1}^{n}\sum_{j=1}^{m_i} y_{i,j}E_{i,j} \le E_{\text{tot}}, \quad i=1,\dots,n,
\end{aligned}$$

where $n=3$ denotes the total number of modules within the software system, m_i denotes number of versions available for module i, and $m_1=3$, $m_2=3$, $m_3=2$. $R_{i,j,h}$ and $E_{i,j,h}$ denote the reliability and the cost of module i with version j in scenario h, respectively, and their nominal values are summarized in Table C.1. It is assumed that $R_{1,3}$, $R_{2,1}$ and $R_{3,2}$ are uncertain and independently obey normal distributions with means of 0.80, 0.90, 0.94 and the same standard deviation of 0.017. w_h denotes the probability associated with scenario h. The reliabilities and parameter values in the s scenarios addressed in the problem can be calculated according to the sampling rule explained in Appendix A. The deterministic parameters take their nominal values for all the scenarios.

The stochastic problem is further reformulated with additional variables for ease of applying McCormick relaxation. These additional variables are defined as follows:

$$
\begin{aligned}
& q_{1,i,h} = \ln(1 - r_{i,h}), \quad q_{2,h} = r_{1,h} r_{2,h}, \quad q_{3,h} = q_{2,h} r_{3,h}, \\
& i = 1, ..., 3, \quad h = 1, ..., s.
\end{aligned}
$$

D The Pump Configuration Problem

The deterministic problem formulation, if expressing all the integer decisions with binary variables and considering a two-level pump network [45] [22], can be written as follows:

$$
\begin{aligned}
\min_{\substack{y^{\text{np}}_{j,i}, y^{\text{ns}}_{k,i}, z_i, \\ P_i, \Delta p_i, \dot{v}_i, x_i, \omega_i}} \quad & \sum_{i=1}^{2} \left[(C_i^{\text{fix}} + C_i P_i) \left(\sum_{j=1}^{2} 2^{j-1} y^{\text{np}}_{j,i} \right) \left(\sum_{k=1}^{2} 2^{k-1} y^{\text{np}}_{k,i} \right) \right] \\
\text{s.t.} \quad & P_i - \alpha_i \left(\frac{\omega_i}{\omega_{\max}} \right)^3 - \beta_i \left(\frac{\omega_i}{\omega_{\max}} \right)^2 \dot{v}_i - \gamma_i \left(\frac{\omega_i}{\omega_{\max}} \right) \dot{v}_i^2 = 0, \\
& \Delta p_i - a_i \left(\frac{\omega_i}{\omega_{\max}} \right)^2 - b_i \left(\frac{\omega_i}{\omega_{\max}} \right) \dot{v}_i - c_i \left(\frac{\omega_i}{\omega_{\max}} \right) \dot{v}_i^2 = 0, \\
& \sum_{i=1}^{3} x_i = 1, \\
& \dot{v}_i \left(\sum_{j=1}^{2} 2^{j-1} y^{\text{np}}_{j,i} \right) - x_i V_{\text{tot}} = 0, \\
& \Delta P_{\text{tot}} z_i - \Delta p_i \left(\sum_{k=1}^{2} 2^{k-1} y^{\text{ns}}_{k,i} \right) = 0, \\
& 0 \leq x_i \leq 1, \quad 0 \leq \dot{v}_i \leq V_{\text{tot}}, \quad 0 \leq \omega_i \leq \omega_{\max}, \\
& 0 \leq P_i \leq P_i^{\max}, \quad 0 \leq \Delta p_i \leq \Delta P_{\text{tot}}, \quad P_i \leq z_i P_i^{\max}, \\
& \Delta p_i \leq z_i \Delta P_{\text{tot}}, \quad \dot{v}_i \leq z_i V_{\text{tot}}, \quad x_i \leq z_i, \\
& \omega_i \leq z_i \omega_{\max}, \quad \sum_{j=1}^{2} 2^{j-1} y^{\text{np}}_{j,i} \leq 3 z_i, \quad \sum_{k=1}^{2} 2^{k-1} y^{\text{np}}_{k,i} \leq 3 z_i, \\
& y^{\text{np}}_{j,i} \in \{0,1\}, \quad y^{\text{ns}}_{k,i} \in \{0,1\}, \quad z_i \in \{0,1\}, \\
& \forall i \in \{1,2\}, \quad \forall j \in \{1,2\}, \quad \forall k \in \{1,2\},
\end{aligned}
\tag{PCP-D}
$$

where index i indicates different levels of the pump network, variable z_i decides whether pump level i is developed or not, $y^{\text{np}}_{j,i}$ decides the number of parallel lines on level i (the

Table D.1 Level specific parameters for the pump configuration problem

Parameter	Pump Level 1	Pump Level 2
C_i^{fix} (FIM) [a]	6329.03	2489.31
C_i (FIM/kW)	1800	1800
α_i	19.9	1.21
β_i	0.161	0.0644
γ_i	-0.000561	-0.000564
a_i	629.0	215.0
b_i	0.696	2.95
c_i	-0.0116	-0.115
$P_i^{\max}$ (kW)	80	25

[a] FIM stands for Finish Markka.

number is $\sum_{j=1}^{2} 2^{j-1} y_{j,i}^{\text{np}}$) and $y_{k,i}^{\text{ns}}$ decides the number of serial lines on level i (the number is $\sum_{k=1}^{2} 2^{k-1} y_{k,i}^{\text{np}}$). The continuous variables include the fraction of total flow going to level i, x_i, the flow rate on each line at level i, $\dot{v}_i$, the rotation speed of pumps on level i, ω_i, the power requirements at level i, P_i, and the pressure rise at level i, Δp_i. The parameters $V_{\text{tot}} = 350\text{m}^3/\text{h}$, $\Delta P_{\text{tot}} = 400\text{kPa}$, $\omega_{\max} = 2950\text{rpm}$. The level specific parameters of the problem are summarized in Table D.1.

Notice that the objective function of Problem (PCP-D) has nonlinear functions of binary variables and it is not separable in continuous and binary variables as in Problem (P). The multiplication of a binary variable and a continuous variable can be modeled alternatively by introducing additional variables with a set of linear constraints [48]. This approach is adopted in this paper, i.e., additional binary variables $y_{j,k,i}$ and continuous variables $P_{j,k,i}^{\text{y}}$, $\dot{v}_{j,i}^{\text{y}}$, $\Delta p_{k,i}^{\text{y}}$ are introduced such that:

$$
\begin{aligned}
& y_{j,k,i} = y_{j,i}^{\text{np}} y_{k,i}^{\text{ns}}, \\
& P_{j,k,i}^{\text{y}} = P_i y_{j,k,i}, \\
& \dot{v}_{j,i}^{\text{y}} = \dot{v}_i y_{j,i}^{\text{np}}, \\
& \Delta p_{k,i}^{\text{y}} = \Delta p_i y_{k,i}^{\text{ns}}, \\
& \forall i \in \{1,2,3\},\ \forall j \in \{1,2\},\ \forall k \in \{1,2\},
\end{aligned}
$$

which are equivalent to

$$
\begin{aligned}
& y_{j,k,i} \le y^{\mathrm{np}}_{j,i}, \quad y_{j,k,i} \le y^{\mathrm{ns}}_{k,i}, \quad y_{j,k,i} \ge y^{\mathrm{np}}_{j,i} + y^{\mathrm{ns}}_{k,i} - 1, \quad y_{j,k,i} \ge 0, \\
& P^{\mathrm{y}}_{j,k,i} \le P^{\max}_i y_{j,k,i}, \quad P^{\mathrm{y}}_{j,k,i} \ge 0, \quad P^{\mathrm{y}}_{j,k,i} \le P_i, \quad P^{\mathrm{y}}_{j,k,i} \ge P_i - P^{\max}_i (1 - y_{j,k,i}), \\
& \dot{v}^{\mathrm{y}}_{j,i} \le V_{\mathrm{tot}} y^{\mathrm{np}}_{j,i}, \quad \dot{v}^{\mathrm{y}}_{j,i} \ge 0, \quad \dot{v}^{\mathrm{y}}_{j,i} \le \dot{v}_i, \quad \dot{v}^{\mathrm{y}}_{j,i} \ge \dot{v}_i - V_{\mathrm{tot}} (1 - y^{\mathrm{np}}_{j,i}), \\
& \Delta p^{\mathrm{y}}_{k,i} \le \Delta P_{\mathrm{tot}} y^{\mathrm{ns}}_{k,i}, \quad \Delta p^{\mathrm{y}}_{k,i} \ge 0, \quad \Delta p^{\mathrm{y}}_{k,i} \le \Delta p_i, \quad \Delta p^{\mathrm{y}}_{k,i} \ge \Delta p_i - \Delta P_{\mathrm{tot}} (1 - y^{\mathrm{ns}}_{k,i}), \\
& \forall i \in \{1,2\}, \, \forall j \in \{1,2\}, \, \forall k \in \{1,2\}.
\end{aligned}
$$

Also, the following constraints can be added into this engineering problem to tighten the feasible set:

$$
\sum_{i=1}^{2} z_i \ge 1, \quad \sum_{j=1}^{2} 2^{j-1} y^{\mathrm{np}}_{j,i} \ge z_i, \quad \sum_{k=1}^{2} 2^{k-1} y^{\mathrm{ns}}_{k,i} \ge z_i, \quad \forall i \in \{1,2\},
$$

which require at least 1 pump level to be developed and at least 1 pump exists in a pump level if this pump level is developed.

According to the proposed changes for the deterministic problem (PCP-D), the stochastic problem formulation can be written as:

$$
\begin{aligned}
&\min_{\substack{y^{\mathrm{np}}_{j,i},y^{\mathrm{ns}}_{k,i},z_i,y_{j,k,i},\\ P_{i,h},\Delta p_{i,h},\dot{v}_{i,h},x_{i,h},\omega_{i,h}\\ P^{\mathrm{y}}_{j,k,i,h},\Delta p^{\mathrm{y}}_{k,i,h},\dot{v}^{\mathrm{y}}_{j,i,h}}} \quad \sum_{h=1}^{s} w_h \left[\sum_{i=1}^{2}\left(C_i^{\mathrm{fix}}\sum_{j=1}^{2}\sum_{k=1}^{2}2^{j+k-2}y_{j,k,i}+C_i\sum_{j=1}^{2}\sum_{k=1}^{2}2^{j+k-2}P^{\mathrm{y}}_{j,k,i,h}\right)\right]\\
&\text{s.t. } P_{i,h}-\alpha_{i,h}\left(\frac{\omega_{i,h}}{\omega_{\max}}\right)^3-\beta_i\left(\frac{\omega_{i,h}}{\omega_{\max}}\right)^2\dot{v}_{i,h}-\gamma_i\left(\frac{\omega_{i,h}}{\omega_{\max}}\right)\dot{v}_{i,h}^2=0,\\
&\quad \Delta p_{i,h}-a_i\left(\frac{\omega_{i,h}}{\omega_{\max}}\right)^2-b_i\left(\frac{\omega_{i,h}}{\omega_{\max}}\right)\dot{v}_{i,h}-c_i\left(\frac{\omega_{i,h}}{\omega_{\max}}\right)\dot{v}_{i,h}^2=0,\\
&\quad \sum_{i=1}^{3}x_{i,h}=1,\\
&\quad \sum_{j=1}^{2}2^{j-1}\dot{v}^{\mathrm{y}}_{j,i,h}-x_{i,h}V_{\mathrm{tot}}=0,\\
&\quad \Delta P_{\mathrm{tot}}z_i-\sum_{k=1}^{2}2^{k-1}\Delta p^{\mathrm{y}}_{k,i,h}=0,\\
&\quad 0\le x_{i,h}\le 1,\quad 0\le \dot{v}_{i,h}\le V_{\mathrm{tot}},\quad 0\le\omega_{i,h}\le\omega_{\max},\\
&\quad 0\le P_{i,h}\le P_i^{\max},\quad 0\le\Delta p_{i,h}\le\Delta P_{\mathrm{tot}},\quad P_{i,h}\le z_iP_i^{\max},\\
&\quad \Delta p_{i,h}\le z_i\Delta P_{\mathrm{tot}},\quad \dot{v}_{i,h}\le z_iV_{\mathrm{tot}},\quad x_{i,h}\le z_i,\\
&\quad \omega_{i,h}\le z_i\omega_{\max},\quad \sum_{j=1}^{2}2^{j-1}y^{\mathrm{np}}_{j,i}\le 3z_i,\quad \sum_{k=1}^{2}2^{k-1}y^{\mathrm{np}}_{k,i}\le 3z_i,\\
&\quad \sum_{i=1}^{2}z_i\ge 1,\quad \sum_{j=1}^{2}2^{j-1}y^{\mathrm{np}}_{j,i}\ge z_i,\quad \sum_{k=1}^{2}2^{k-1}y^{\mathrm{ns}}_{k,i}\ge z_i,\\
&\quad y_{j,k,i}\le y^{\mathrm{np}}_{j,i},\quad y_{j,k,i}\le y^{\mathrm{ns}}_{k,i},\quad y_{j,k,i}\ge y^{\mathrm{np}}_{j,i}+y^{\mathrm{ns}}_{k,i}-1,\\
&\quad 0\le P^{\mathrm{y}}_{j,k,i,h}\le P_i^{\max}y_{j,k,i},\quad P_{i,h}-P_i^{\max}(1-y_{j,k,i})\le P^{\mathrm{y}}_{j,k,i,h}\le P_{i,h},\\
&\quad 0\le\dot{v}^{\mathrm{y}}_{j,i,h}\le V_{\mathrm{tot}}y^{\mathrm{np}}_{j,i},\quad \dot{v}_{i,h}-V_{\mathrm{tot}}(1-y^{\mathrm{np}}_{j,i})\le\dot{v}^{\mathrm{y}}_{j,i,h}\le\dot{v}_{i,h},\\
&\quad 0\le\Delta p^{\mathrm{y}}_{k,i,h}\le\Delta P_{\mathrm{tot}}y^{\mathrm{ns}}_{k,i},\quad \Delta p_{i,h}-\Delta P_{\mathrm{tot}}(1-y^{\mathrm{ns}}_{k,i})\le\Delta p^{\mathrm{y}}_{k,i,h}\le\Delta p_{i,h},\\
&\quad y^{\mathrm{np}}_{j,i}\in\{0,1\},\quad y^{\mathrm{ns}}_{k,i}\in\{0,1\},\quad z_i\in\{0,1\},\quad y_{j,k,i}\in\{0,1\},\\
&\quad \forall i\in\{1,2\},\quad\forall j\in\{1,2\},\quad\forall k\in\{1,2\},\quad\forall h\in\{1,...,s\}.
\end{aligned}
\tag{PCP-S}
$$

It is assumed that parameters a_1, a_2 and α_2 in the pump performance models are uncertain and independently obey normal distributions with means of 629.0, 215.0, 1.21 and standard deviations of 62.90, 21.50, 0.121, respectively. w_h denotes the probability associated with scenario h. The probabilities and values of uncertain parameter values in the s scenarios addressed in the problem can be calculated according to the sampling rule explained in Appendix A. The deterministic parameters take their nominal values for all the scenarios.

The stochastic problem is further reformulated with additional variables for ease of applying McCormick relaxation. These additional variables are defined as follows:

$$q_{1,i,h} = (\dot{v}_{i,h})^2, \quad q_{2,i,h} = (\omega_{i,h})^2, \quad q_{3,i,h} = (\omega_{i,h})^3,$$
$$q_{4,i,h} = q_{2,i,h}\dot{v}_{i,h}, \quad q_{5,i,h} = q_{1,i,h}\omega_{i,h}, \quad q_{6,i,h} = \dot{v}_{i,h}\omega_{i,h},$$
$$i = 1,2, \quad h = 1,...,s.$$

E The Sarawak Gas Production Subsystem Problem

This problem was studied in [47] to optimize the short-term operation of the subsystem. Here the problem is revised into an integrated design and operation problem with additional binary variables that determine whether to develop part of the subsystem or not. Figure E.1 shows the superstructuree of the subsystem, where the solid lines indicate the part of the subsystem that has to be developed due to specific engineering reasons and dashed lines indicate the part of the subsystem that may be developed for the system. Nodes M4, M3 and SE denote three gas fields which have 2, 10 and 2 gas wells, respectively. Node M3P is a gas platform where there exists a compressor to increase pressure of gas flows. Node T is a terminal node for this subsystem where the outlet gas flow must satisfy demands at this node. The deterministic formulation of the problem consists of a mass balance model, a pressure model, a compressor model and an economic objective.

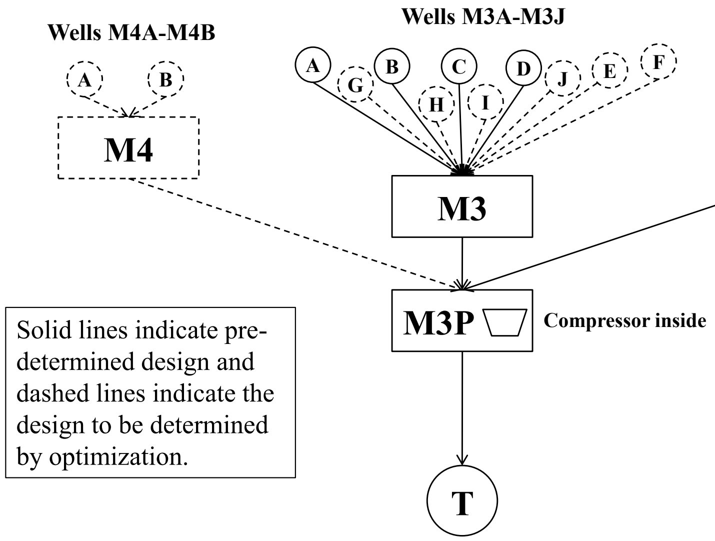

Figure E.1 The superstructure of the Sarawak Gas Production Subsystem.

The mass balance model is as follows:

$$
\begin{aligned}
&\sum_{i\in\Omega_j} f_{i,j} = f_{j,\mathrm{M3P}}, \quad \forall j \in \{\mathrm{M4},\mathrm{M3},\mathrm{SE}\},\\
&\sum_{j\in\{\mathrm{M4},\mathrm{M3},\mathrm{SE}\}} f_{j,\mathrm{M3P}} - f^{\mathrm{loss}} = f_{\mathrm{M3P,T}}, \quad f^{\mathrm{loss}} \geq 0,\\
&D^{\mathrm{lo}} \leq f_{\mathrm{M3P,T}} \leq D^{\mathrm{up}},\\
&y^{\mathrm{P}}_{i,j} f^{\mathrm{lo}}_{i,j} \leq f_{i,j} \leq y^{\mathrm{P}}_{i,j} f^{\mathrm{up}}_{i,j}, \quad \forall j \in \{\mathrm{M3P},\mathrm{T}\},\, \forall i \in \Omega_j,\\
&y^{\mathrm{P}}_{i,j} = y^{\mathrm{N}}_i, \quad y^{\mathrm{P}}_{i,j} \leq y^{\mathrm{N}}_j, \quad \forall j \in \{\mathrm{M4},\mathrm{M3},\mathrm{SE}\},\, \forall i \in \Omega_j,\\
&\sum_{i\in\Omega_j} y^{\mathrm{P}}_{i,j} \geq y^{\mathrm{N}}_j, \quad y^{\mathrm{N}}_j = y^{\mathrm{P}}_{j,\mathrm{M3P}}, \quad \forall j \in \{\mathrm{M4},\mathrm{M3},\mathrm{SE}\},\, \forall i \in \Omega_j,\\
&y^{\mathrm{P}}_{i,j} \in \{0,1\},\, \forall (i,j) \in \Theta^{\mathrm{P}}, \quad y^{\mathrm{N}}_i \in \{0,1\},\, \forall i \in \Theta^{\mathrm{N}},
\end{aligned}
\tag{SGPS-MB}
$$

where $f_{i,j}$ denotes the gas flow from a node i to a node j and $f^{\mathrm{lo}}_{i,j}$ and $f^{\mathrm{up}}_{i,j}$ denote the relevant bounds due to pipeline capacities, D^{lo} denotes the minimum demand rate that has to be satisfied at node T and D^{up} denotes the maximum demand rate at node T (which is also the maximum gas flow rate that can enter node T), f^{loss} denotes the loss of gas at node M3P that is used to generate power for the compressor, $y^{\mathrm{P}}_{i,j}$ represents the decision on whether to develop a pipeline between node i and j, y^{N}_i represents the decision on whether to develop node i, and the constraints on the binary variables enforce relevant topology restrictions. Set Ω_j contains indices of the nodes which the gas flow entering node j can come from, set Θ^{P} contains pairs of indices that indicate all the possible pipelines in the system, and set Θ^{N} contains indices of the possible nodes in the system. Nominal values of the relevant parameters can be found in Table E.1.

Table E.1 Some parameters for the SGPS model

Parameter	Unit	Nominal Value	Note
$f_{i,j}^{\mathrm{lo}}$	Mscfd [a]	0	$\forall (i,j) \in \Theta^{\mathrm{P}}$
$f_{i,j}^{\mathrm{up}}$	Mscfd	3000	$\forall (i,j) \in \Theta^{\mathrm{P}}$
D^{lo}	Mscfd	600	
D^{up}	Mscfd	850	
$P_{\mathrm{c,M3P}}^{\mathrm{up}}$	bar	78.36	
$P_{\mathrm{c,M3P}}^{\mathrm{lo}}$	bar	1	
$P_{\mathrm{M3P}}^{\mathrm{up}}$	bar	200	
$P_{\mathrm{T}}^{\mathrm{lo}}$	bar	30	
$P_{\mathrm{T}}^{\mathrm{up}}$	bar	80	
$P_{t,i}^{\mathrm{lo}}$	bar	1	$\forall i \in \Omega_j,\ \forall j \in \{\mathrm{M4,M3,SE}\}$
$P_{t,i}^{\mathrm{up}}$	bar	$\pi_{r,i}$	$\forall i \in \Omega_j,\ \forall j \in \{\mathrm{M4,M3,SE}\}$
			(See Table E.2 for $\pi_{r,i}$)
κ	$\mathrm{bar}^2 \cdot \mathrm{day}^2/\mathrm{hm}^6$ [b]	2.46	
ρ	$(\mathrm{hm}^3/\mathrm{day}^2)$/Mscfd	0.0283168	
ζ	MW/$(\mathrm{hm}^3/\mathrm{day})$	253.96	
ω	MW/$(\mathrm{hm}^3/\mathrm{day})$	5.236	
ν		0.333	
C_0	Million \$	3000	
$C_{1,i}$	Million \$	10	$\forall i \in \Theta^{\mathrm{N}} \setminus \{\mathrm{M3P}\}$
		500	For $i = \mathrm{M3P}$
$C_{2,i,j}$	Million \$	16	For $(i,j) = (\mathrm{M4,M3P})$
		50	For $(i,j) = (\mathrm{SE,M3P})$
		40	For $(i,j) = (\mathrm{M3P,T})$
		0	For other (i,j) in Θ^{P}
C_3	\$/Mscf	4480	
T	year	25	
σ	%	12	

[a] Mscfd stands for million standard cubic feet per day.
[b] hm stands for 10^2m.

The pressure model can be written as:

$$
\begin{aligned}
&\alpha_i \rho f_{i,j} + \beta_i \rho^2 f_{i,j}^2 = \pi_{r,i}^2 - P_{b,i}^2, \quad P_{b,i} \geq 0, \quad \forall j \in \{\text{M4}, \text{M3}, \text{SE}\},\ \forall i \in \Omega_j,\\
&\theta_i \rho^2 f_{i,j}^2 = P_{b,i}^2 - \lambda_i P_{t,i}^2, \quad P_{t,i} \geq 0, \quad \forall j \in \{\text{M4}, \text{M3}, \text{SE}\},\ \forall i \in \Omega_j,\\
&P_{\text{c,M3P}}^{\text{up}} + y_{j,\text{M3P}}^{\text{P}} (P_{t,i} - P_{\text{c,M3P}}^{\text{up}}) \geq P_{\text{c,M3P}}, \quad \forall j \in \{\text{M4}, \text{M3}, \text{SE}\},\ \forall i \in \Omega_j,\\
&P_{\text{c,M3P}}^{\text{lo}} \leq P_{\text{c,M3P}}, \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(SGPS-PM)}\\
&P_{\text{c,M3P}}^{\text{lo}} \leq P_{\text{M3P}} \leq P_{\text{M3P}}^{\text{up}},\\
&P_{\text{M3P}}^2 - P_{\text{T}}^2 = \kappa \rho^2 f_{\text{M3P,T}}^2,\\
&P_{\text{T}}^{\text{lo}} \leq P_{\text{T}} \leq P_{\text{T}}^{\text{up}},
\end{aligned}
$$

where $P_{b,i}$ and $P_{t,i}$ denote the well bottom pressure and well head pressure of well i, respectively, $P_{c,\text{M3P}}$ and P_{M3P} denote the pressures of inlet and outlet gas flow of compressor at M3P, respectively, P_{T} denotes the pressure at node T, parameter ρ converts unit of flow from Mscfd to hm^3/day. Nominal values of the parameters in the well performance models can be found in Table E.2, and nominal values of other relevant parameters can be found in Table E.1. Notice that model (SGPS-PM) contains terms $y_{j,\text{M3P}}^{\text{P}} P_{t,i}$, which are not separable in integer and continuous variables. To render a separable representation for these terms, define new variables $P_{t,i,j}^{\text{y}}$ such that

$$
P_{t,i,j}^{\text{y}} = y_{j,\text{M3P}}^{\text{P}} P_{t,i}, \quad \forall j \in \{\text{M4}, \text{M3}, \text{SE}\},\ \forall i \in \Omega_j,
$$

which is equivalent to

$$
\begin{aligned}
&P_{t,i,j}^{\text{y}} \leq y_{j,\text{M3P}}^{\text{P}} (P_{t,i}^{\text{up}} - P_{\text{c,M3P}}^{\text{up}}),\ \forall j \in \{\text{M4}, \text{M3}, \text{SE}\},\ \forall i \in \Omega_j,\\
&P_{t,i,j}^{\text{y}} \geq y_{j,\text{M3P}}^{\text{P}} (P_{t,i}^{\text{lo}} - P_{\text{c,M3P}}^{\text{up}}),\ \forall j \in \{\text{M4}, \text{M3}, \text{SE}\},\ \forall i \in \Omega_j,\\
&P_{t,i,j}^{\text{y}} \leq P_{t,i} - P_{t,i}^{\text{lo}} + y_{j,\text{M3P}}^{\text{P}} (P_{t,i}^{\text{lo}} - P_{\text{c,M3P}}^{\text{up}}),\ \forall j \in \{\text{M4}, \text{M3}, \text{SE}\},\ \forall i \in \Omega_j,\\
&P_{t,i,j}^{\text{y}} \geq P_{t,i} - P_{t,i}^{\text{up}} + y_{j,\text{M3P}}^{\text{P}} (P_{t,i}^{\text{up}} - P_{\text{c,M3P}}^{\text{up}}),\ \forall j \in \{\text{M4}, \text{M3}, \text{SE}\},\ \forall i \in \Omega_j.
\end{aligned}
$$

Table E.2 Well performance model parameters

	$\pi_{r,i}$	α_i	β_i	λ_i	θ_i
	bar	$\mathrm{bar}^2 \cdot \mathrm{d}/\mathrm{hm}^3$	$\mathrm{bar}^2 \cdot \mathrm{d}^2/\mathrm{hm}^6$		$\mathrm{bar}^2 \cdot \mathrm{d}^2/\mathrm{hm}^6$
M3A	69.33	1.888×10^{-1}	3.445×10^{-4}	1.482	1.090×10^{3}
M3B	82.08	1.662×10^{-1}	3.691×10^{-4}	1.621	1.215×10^{3}
M3C	78.36	1.816×10^{-1}	3.481×10^{-4}	1.539	1.048×10^{3}
M3D	73.13	1.657×10^{-1}	3.695×10^{-4}	1.53	1.076×10^{3}
M3E	79.77	1.627×10^{-1}	3.159×10^{-4}	1.642	1.209×10^{3}
M3F	80.46	1.818×10^{-1}	3.815×10^{-4}	1.446	1.160×10^{3}
M3G	82.30	1.825×10^{-1}	3.513×10^{-4}	1.663	1.056×10^{3}
M3H	76.71	1.749×10^{-1}	3.555×10^{-4}	1.617	1.193×10^{3}
M3I	79.49	1.824×10^{-1}	3.375×10^{-4}	1.621	1.033×10^{3}
M3J	72.10	1.577×10^{-1}	3.502×10^{-4}	1.673	1.148×10^{3}
M4A	75.69	1.724×10^{-1}	3.474×10^{-4}	1.573	1.143×10^{3}
M4B	81.38	1.753×10^{-1}	3.316×10^{-4}	1.484	1.219×10^{3}
SEA	153.40	1.050	4.298×10^{-3}	3.515	3.853×10^{2}
SEB	141.22	1.154	4.566×10^{-3}	3.813	3.845×10^{2}

The compressor model can be written as:

$$
\begin{aligned}
&W = \omega\rho f_{\mathrm{M3P},T}(P^{\mathrm{ratio}} - 1),\\
&\ln P^{\mathrm{ratio}} = \nu(\ln P_{\mathrm{M3P}} - \ln P_{\mathrm{c,M3P}}),\\
&W = \zeta\rho f^{\mathrm{loss}},
\end{aligned}
\tag{SGPS-CM}
$$

where the first two equations describes the compressor performance and the last equation represents that the power for the compressor comes from burning the gas. The composition of the burned gas is assumed to be the composition of gas from gas well M3C, then the coefficient ζ can be calculated according to the heating value of each component of the gas [47] and by assuming 40% of the heat is converted into electric power. Nominal values of the parameters in model (SGPS-CM) can be found in Table E.1.

The economic objective to be optimized is the following net present value:

$$
-C_0 - \sum_{i\in\Theta^{\mathrm{N}}} C_{1,i} y_i^{\mathrm{N}} - \sum_{(i,j)\in\Theta^{\mathrm{P}}} C_{2,i,j} y_{i,j}^{\mathrm{P}} + \sum_{t=1}^{T} 365 C_3 \frac{f_{\mathrm{M3P,T}}}{(1+\sigma)^t} \tag{SGPS-NPV}
$$

where $C_{1,i}$ denotes the investment cost of node i, $C_{2,i,j}$ denotes the investment cost of pipeline connecting nodes i and j, C_3 denotes the gas price, T denotes the lifespan of the subsystem and σ denotes the discount rate during the lifespan, C_0 denotes other costs associated with the subsystem (e.g., the downstream facilities to process the products). Nominal values of the parameters are shown in Table E.1.

According to the deterministic problem formulation, Equations (SGPS-MB), (SGPS-PM), (SGPS-CM) and (SGPS-NPV), a stochstic problem formulation can be written as:

$$
\begin{aligned}
\max_{\substack{y_i^{\mathrm{N}},y_{i,j}^{\mathrm{P}},f_{i,j,h},f_h^{\mathrm{loss}},\\ P_{b,i,h},P_{t,i,h},P_h^{\mathrm{ratio}},\\ P_{t,i,j,h}^{\mathrm{y}},P_{\mathrm{c,M3P},h},P_{\mathrm{M3P},h},\\ P_{\mathrm{T},h},W_h}} \quad & \sum_{h=1}^{s} w_h \left[-C_0 - \sum_{i\in\Theta^{\mathrm{N}}} C_{1,i} y_i^{\mathrm{N}} - \sum_{(i,j)\in\Theta^{\mathrm{P}}} C_{2,i,j} y_{i,j}^{\mathrm{P}} + \sum_{t=1}^{T} 365 C_{3,h} \frac{f_{\mathrm{M3P,T},h}}{(1+\sigma)^t} \right] \\
\text{s.t.} \ & \sum_{i\in\Omega_j} f_{i,j,h} = f_{j,\mathrm{M3P},h}, \quad \forall j \in \{\mathrm{M4,M3,SE}\}, \\
& \sum_{j\in\{\mathrm{M4,M3,SE}\}} f_{j,\mathrm{M3P},h} - f_h^{\mathrm{loss}} = f_{\mathrm{M3P,T},h}, \quad f_h^{\mathrm{loss}} \geq 0, \\
& y_{i,j}^{\mathrm{P}} f_{i,j}^{\mathrm{lo}} \leq f_{i,j,h} \leq y_{i,j}^{\mathrm{P}} f_{i,j}^{\mathrm{up}}, \quad \forall j \in \{\mathrm{M3P,T}\}, \ \forall i \in \Omega_j, \\
& D^{\mathrm{lo}} \leq f_{\mathrm{M3P,T},h} \leq D_h^{\mathrm{up}}, \\
& P_{\mathrm{c,M3P}}^{\mathrm{lo}} \leq P_{\mathrm{c,M3P},h}, \quad P_{\mathrm{c,M3P}}^{\mathrm{lo}} \leq P_{\mathrm{M3P},h} \leq P_{\mathrm{M3P}}^{\mathrm{up}}, \\
& P_{\mathrm{M3P},h}^2 - P_{\mathrm{T},h}^2 = \kappa_h \rho^2 f_{\mathrm{M3P,T},h}^2, \quad P_{\mathrm{T}}^{\mathrm{lo}} \leq P_{\mathrm{T},h} \leq P_{\mathrm{T}}^{\mathrm{up}}, \\
& W_h = \omega \rho f_{\mathrm{M3P,T},h} (P_h^{\mathrm{ratio}} - 1), \quad W_h = \zeta \rho f_h^{\mathrm{loss}}, \\
& \ln P_h^{\mathrm{ratio}} = \nu (\ln P_{\mathrm{M3P},h} - \ln P_{\mathrm{c,M3P},h}), \\
& \left.
\begin{aligned}
& \alpha_i \rho f_{i,j,h} + \beta_i \rho^2 f_{i,j,h}^2 = \pi_{r,i}^2 - P_{b,i,h}^2, \\
& \theta_i \rho^2 f_{i,j,h}^2 = P_{b,i,h}^2 - \lambda_i P_{t,i,h}^2, \\
& P_{b,i,h} \geq 0, \quad P_{t,i,h} \geq 0, \\
& P_{\mathrm{c,M3P}}^{\mathrm{up}} + P_{t,i,j,h}^{\mathrm{y}} - y_{j,\mathrm{M3P}}^{\mathrm{P}} P_{\mathrm{c,M3P},h}^{\mathrm{up}} \geq P_{\mathrm{c,M3P}}, \\
& P_{t,i,j,h}^{\mathrm{y}} \leq y_{j,\mathrm{M3P}}^{\mathrm{P}} (P_{t,i}^{\mathrm{up}} - P_{\mathrm{c,M3P}}^{\mathrm{up}}), \\
& P_{t,i,j,h}^{\mathrm{y}} \geq y_{j,\mathrm{M3P}}^{\mathrm{P}} (P_{t,i}^{\mathrm{lo}} - P_{\mathrm{c,M3P}}^{\mathrm{up}}), \\
& P_{t,i,j,h}^{\mathrm{y}} \leq P_{t,i,h} - P_{t,i}^{\mathrm{lo}} + y_{j,\mathrm{M3P}}^{\mathrm{P}} (P_{t,i}^{\mathrm{lo}} - P_{\mathrm{c,M3P}}^{\mathrm{up}}), \\
& P_{t,i,j,h}^{\mathrm{y}} \geq P_{t,i,h} - P_{t,i}^{\mathrm{up}} + y_{j,\mathrm{M3P}}^{\mathrm{P}} (P_{t,i}^{\mathrm{up}} - P_{\mathrm{c,M3P}}^{\mathrm{up}}), \\
& y_{i,j}^{\mathrm{P}} = y_i^{\mathrm{N}}, \ y_{i,j}^{\mathrm{P}} \leq y_j^{\mathrm{N}}, \ \textstyle\sum_{i\in\Omega_j} y_{i,j}^{\mathrm{P}} \geq y_j^{\mathrm{N}}, \ y_j^{\mathrm{N}} = y_{j,\mathrm{M3P}}^{\mathrm{P}},
\end{aligned}
\right\} \quad
\begin{aligned}
& \forall j \in \{\mathrm{M4,M3,SE}\}, \\
& \forall i \in \Omega_j,
\end{aligned} \\
& y_{i,j}^{\mathrm{P}} \in \{0,1\}, \ \forall (i,j) \in \Theta^{\mathrm{P}}, \quad y_i^{\mathrm{N}} \in \{0,1\}, \ \forall i \in \Theta^{\mathrm{N}}.
\end{aligned}
\tag{SGPS-S}
$$

It is assumed that parameters D^{up}, κ and C_3 are uncertain and independently obey nominal distributions with mean of 850 Mscfd, 2.46 $\mathrm{bar}^2 \cdot \mathrm{day}^2/\mathrm{hm}^6$, 4480 \$/Mscf, and standard deviations of 50 Mscfd, 0.083 $\mathrm{bar}^2 \cdot \mathrm{day}^2/\mathrm{hm}^6$, 149.3 \$/Mscf, respectively. w_h denotes the probability associated with scenario h. The probabilities and the uncertain parameter values in the s scenarios addressed in the problem can be calculated according to the sampling rule explained in Appendix A. The deterministic parameters take their nominal values for all the scenarios.

Again, the stochastic problem is further reformulated with additional variables, which are defined as follows:

$$
\begin{aligned}
& q_{1,i,j,h} = f_{i,j,h}^2, \quad q_{2,i,h} = P_{b,i,h}^2, \quad q_{3,i,h} = P_{t,i,h}^2, \\
& q_{4,h} = P_{\mathrm{M3P},h}^2, \quad q_{5,h} = P_{\mathrm{T},h}^2, \quad q_{6,h} = f_{\mathrm{M3P,T},h}^2, \quad q_{7,h} = f_{\mathrm{M3P,T},h} P_h^{\mathrm{ratio}}, \\
& q_{8,h} = \ln P_h^{\mathrm{ratio}}, \quad q_{9,h} = \ln P_{\mathrm{M3P},h}, \quad q_{10,h} = \ln P_{\mathrm{c,M3P},h}, \\
& \forall j \in \{\mathrm{M4}, \mathrm{M3}, \mathrm{SE}\}, \; \forall i \in \Omega_j, \; h = 1, \ldots, s.
\end{aligned}
$$

```
*================================================================================
*============ The GAMS file for the software reliability problem ===============
*================================================================================

SET
nx 'Set of continuous variables' /1*8/,
ny 'Set of binary variables' /1*8/,
* The relationship between h and subh is h=subh^3
h Set of sencarios /1*27/,
subh Set of uncertain scenarios for each uncertain parameter /1*3/;

alias(subh, subh2);
alias(subh, subh3);


PARAMETERS
Prob_h /0/,

Prob(h) /1 0/,

y_coeff_nominal(ny)
/1  0.1
 2  0.2
 3  0.15
 4  0.05
 5  0.2
 6  0.15
 7  0.02
 8  0.06/,

y_coeff_ave(ny)
/1  0.1
 2  0.2
 3  0.2
 4  0.1
 5  0.2
 6  0.15
 7  0.02
 8  0.06/,

y_coeff_range(ny)
/1  0/,

y_coeff(ny,h)
/1  .1  0/,

y_coeff_h(ny)
/1  0/,

subProb(subh)
/1  0/;



*=========== Generate scenarios for three uncertain parameters ============

y_coeff_range('3')=0.1;
y_coeff_range('8')=0.1;
y_coeff_range('4')=0.1;
y_coeff(ny,h)=y_coeff_nominal(ny);
```

```
subProb(subh)$(card(subh)>1 and ord(subh)=1) = errorf(-3+6/card(subh));
subProb(subh)$(card(subh)>1 and ord(subh)>1 and ord(subh)<card(subh))
     = errorf(-3+ord(subh)*6/card(subh))-errorf(-3+(ord(subh)-1)*6/card(subh));
subProb(subh)$(card(subh)>1 and ord(subh) = card(subh))
     = 1 - errorf(-3+(card(subh)-1)*6/card(subh));

if (card(h)=1,
        Prob(h)=1;
        y_coeff(ny,h)=y_coeff_nominal(ny);
else
    loop(subh,
      loop(subh2,
        loop(subh3,
               Prob(h)$(ord(h)=ord(subh)+(ord(subh2)-1)*card(subh)
                               +(ord(subh3)-1)*card(subh)*card(subh2))
                      = subProb(subh)*subProb(subh2)*subProb(subh3);

               y_coeff('3',h)$(ord(h)=ord(subh)+(ord(subh2)-1)*card(subh)
                                      +(ord(subh3)-1)*card(subh)*card(subh2))
                      = y_coeff_ave('3')-y_coeff_range('3')/2
                        +y_coeff_range('3')/card(subh)/2
                        +(ord(subh)-1)*y_coeff_range('3')/card(subh);

               y_coeff('8',h)$(ord(h)=ord(subh)+(ord(subh2)-1)*card(subh)
                                      +(ord(subh3)-1)*card(subh)*card(subh2))
                      = y_coeff_ave('8')-y_coeff_range('8')/2
                        +y_coeff_range('8')/card(subh2)/2
                        +(ord(subh2)-1)*y_coeff_range('8')/card(subh2);

               y_coeff('4',h)$(ord(h)=ord(subh)+(ord(subh2)-1)*card(subh)
                                      +(ord(subh3)-1)*card(subh)*card(subh2))
                      = y_coeff_ave('4')-y_coeff_range('4')/2
                        +y_coeff_range('4')/card(subh3)/2
                        +(ord(subh3)-1)*y_coeff_range('4')/card(subh3);

                 );
               );
             );
    );



BINARY VARIABLES
y(ny);

VARIABLES
x(nx,h), objvalue;

x.up('1',h)=1-y_coeff('1',h)*y_coeff('2',h)*y_coeff('3',h);
x.lo('1',h)=0;
x.up('2',h)=1-y_coeff('4',h)*y_coeff('5',h)*y_coeff('6',h);
x.lo('2',h)=0;
x.up('3',h)=1-y_coeff('7',h)*y_coeff('8',h);
x.lo('3',h)=0;

x.up('4',h)=x.up('1',h)*x.up('2',h);
x.lo('4',h)=x.lo('1',h)*x.lo('2',h);
x.up('5',h)=x.up('4',h)*x.up('3',h);
x.lo('5',h)=x.lo('4',h)*x.lo('3',h);
```

```
x.up('6',h)=log(1-x.lo('1',h));
x.lo('6',h)=log(1-x.up('1',h));
x.up('7',h)=log(1-x.lo('2',h));
x.lo('7',h)=log(1-x.up('2',h));
x.up('8',h)=log(1-x.lo('3',h));
x.lo('8',h)=log(1-x.up('3',h));


EQUATIONS
         Objective
         newequ1(h)
         newequ2(h)
         newequ3(h)
         newequ4(h)
         newequ5(h)
         equ1(h)
         equ2(h)
         equ3(h)
         inequ1
         inequ2
         inequ3
         inequ4 ;

Objective.. objvalue =e= 100*sum(h, -x('5',h)*Prob(h));

newequ1(h).. x('4',h) =e= x('1',h)*x('2',h);
newequ2(h).. x('5',h) =e= x('4',h)*x('3',h);
newequ3(h).. x('6',h) =e= log(1-x('1',h));
newequ4(h).. x('7',h) =e= log(1-x('2',h));
newequ5(h).. x('8',h) =e= log(1-x('3',h));

equ1(h).. LOG(y_coeff('1',h))*y('1')+LOG(y_coeff('2',h))*y('2')
                              +LOG(y_coeff('3',h))*y('3')-x('6',h) =e= 0;

equ2(h).. LOG(y_coeff('4',h))*y('4')+LOG(y_coeff('5',h))*y('5')
                              +LOG(y_coeff('6',h))*y('6')-x('7',h) =e= 0;

equ3(h).. LOG(y_coeff('7',h))*y('7')+LOG(y_coeff('8',h))*y('8')-x('8',h) =e= 0;

inequ1.. -y('1')-y('2')-y('3') =l= -1;

inequ2.. -y('4')-y('5')-y('6') =l= -1;

inequ3.. -y('7')-y('8') =l= -1;

inequ4.. 3*y('1')+y('2')+2*y('3')+3*y('4')+2*y('5')+y('6')+3*y('7')+2*y('8') =l= 10;


MODEL Software /ALL/;
Software.OPTFILE=1;
Software.OPTCA=1e-3;
Software.OPTCR=1e-3;
Software.reslim = 1e9;
Software.iterlim = 1e5;

OPTION MINLP=BARON;
SOLVE Software USING MINLP MINIMIZING objvalue;
```

```
*==============================================================================
*========== The GAMS file for the pump network configuration problem ===========
*==============================================================================

SETS
superi Superset of levels /1*3/,
i(superi) Set of levels /1*2/,
k For representing general integer variables /1*2/,
* The relationship between h and subh is h=subh^3
h Set of sencarios /1*27/,
subh Set of uncertain scenarios for each uncertain parameter /1*3/;

alias(k,k2);
alias(subh, subh2);
alias(subh, subh3);


PARAMETERS
wmax Maximum rotation speed
/2950/,
Vtot Total volumetric flowrate
/350/,
dPtot Total pressure rise
/400/,
Prob(h) Probability for each scenario
/1  0/,

Prob_h,

Pmax(superi) Maximum power output
        /'1'    80
         '2'    25
         '3'    45/,

C(superi) Fixed cost of pump
        /'1'    6329.03
         '2'    2489.31
         '3'    3270.27/,

Cd(superi) Operating cost coefficient
        /'1'    1800
         '2'    1800
         '3'    1800/,

alpha_nominal(superi)
/1 19.9
 2 1.21
 3 6.52/,

alpha_h(superi)
/1  0/,

alpha_range(superi)
/1  0/,

alpha(superi, h)
/1  .1  0/,

beta(superi)
/1 0.161
```

```
 2 0.0644
 3 0.102/,

gamma(superi)
/1 -0.000561
 2 -0.000564
 3 -0.000232/,

aa_nominal(superi)
/1 629
 2 215
 3 361/,

aa_h(superi)
/1  0/,

aa_range(superi)
/1  0/,

aa(superi, h)
/1  .1  0/,

bb_nominal(superi)
/1 0.696
 2 2.95
 3 0.530/,

bb_h(superi)
/1  0/,

bb_range(superi)
/1  0/,

bb(superi, h)
/1  .1  0/,

cc_nominal(superi)
/1 -0.0116
 2 -0.115
 3 -0.00946/,

cc_h(superi)
/1  0/,

cc_range(superi)
/1  0/,

cc(superi, h)
/1  .1  0/

subProb(subh)
/1  0/
;

alpha(superi, h)=alpha(superi, h)/Pmax(superi);
beta(superi)=beta(superi)/Pmax(superi)*Vtot;
gamma(superi)=gamma(superi)/Pmax(superi)*sqr(Vtot);
aa(superi, h)=aa(superi, h)/dPtot;
bb(superi, h)=bb(superi, h)/dPtot*Vtot;
cc(superi, h)=cc(superi, h)/dPtot*sqr(Vtot);
```

```
C(superi)=C(superi)/1000;
Cd(superi)=Cd(superi)/1000;




*=========== Generate scenarios for three uncertain parameters =============

alpha(superi,h)=alpha_nominal(superi);
alpha_range(superi)=0.6*alpha_nominal(superi);
aa(superi,h)=aa_nominal(superi);
aa_range(superi)=0.6*aa_nominal(superi);
bb(superi,h)=bb_nominal(superi);
cc(superi,h)=cc_nominal(superi);

subProb(subh)$(card(subh)>1 and ord(subh)=1) = errorf(-3+6/card(subh));
subProb(subh)$(card(subh)>1 and ord(subh)>1 and ord(subh)<card(subh))
  = errorf(-3+ord(subh)*6/card(subh)) - errorf(-3+(ord(subh)-1)*6/card(subh));
subProb(subh)$(card(subh)>1 and ord(subh) = card(subh))
  = 1 - errorf(-3+(card(subh)-1)*6/card(subh));

if (card(h)=1,
        Prob(h)=1;
        alpha(superi,h)=alpha_nominal(superi);
        aa(superi,h)=aa_nominal(superi);
else
    loop(subh,
      loop(subh2,
        loop(subh3,
            Prob(h)$(ord(h)=ord(subh)+(ord(subh2)-1)*card(subh)
                            +(ord(subh3)-1)*card(subh)*card(subh2))
                   = subProb(subh)*subProb(subh2)*subProb(subh3);

            aa('1',h)$(ord(h)=ord(subh)+(ord(subh2)-1)*card(subh)
                            +(ord(subh3)-1)*card(subh)*card(subh2))
                   = aa_nominal('1')-aa_range('1')/2+aa_range('1')/card(subh)/2
                     +(ord(subh)-1)*aa_range('1')/card(subh);

            aa('2',h)$(ord(h)=ord(subh)+(ord(subh2)-1)*card(subh)
                            +(ord(subh3)-1)*card(subh)*card(subh2))
                   = aa_nominal('2')-aa_range('2')/2+aa_range('2')/card(subh2)/2
                     +(ord(subh2)-1)*aa_range('2')/card(subh2);

            alpha('2',h)$(ord(h)=ord(subh)+(ord(subh2)-1)*card(subh)
                            +(ord(subh3)-1)*card(subh)*card(subh2))
                   = alpha_nominal('2')-alpha_range('2')/2
                     +alpha_range('2')/card(subh3)/2
                     +(ord(subh3)-1)*alpha_range('2')/card(subh3);
                 );
               );
            );
    );




POSITIVE VARIABLES
         P(i,h)      Power output of pumps on level i
         w(i,h)      Rotation speed for pumps on level i
         dp(i,h)     Pressure rise on level i
```

```
         vdot(i,h)  Flow through pumps on level i
         x(i,h)     Fraction of total flow on level i
         q1(i,h)
         q2(i,h)
         q3(i,h)
         q4(i,h)
         q5(i,h)
         q6(i,h)
         dummyP(k,k2,i,h)
         dummyvdot(k,i,h)
         dummydp(k,i,h)
         y_dummy(k,k2,i)
         ;

P.UP(i,h) = 1;
w.UP(i,h) = 1;
dp.UP(i,h) = 1;
vdot.UP(i,h) = 1;
x.UP(i,h) = 1;
q1.up(i,h) = 1;
q2.up(i,h) = 1;
q3.up(i,h)=1;
q4.up(i,h)=1;
q5.up(i,h)=1;
q6.up(i,h)=1;
dummyP.up(k,k2,i,h)=1;
dummyvdot.up(k,i,h)=1;
dummydp.up(k,i,h)=1;


FREE VARIABLE  objvalue;

BINARY VARIABLES
y_yns(i,k) 'binary variables representing ns'
y_ynp(i,k) 'binary variables representing np'
y_zz(i)    'binary variable representing z'
;



EQUATIONS
         Objective   Objective function
         gP(i,h)     Power output calculation for level i pumps
         gdp(i,h)    Pressure rise calculation for level i
         sumx(h)     Constraint on volume fractions
         gvdot(i,h)  Volume flowrate calculation for pumps on level i
         gdpc(i,h)   Constraints on pressure rise
         lw(i,h)     Logical constraints on w
         lP(i,h)     Logical constraints on P
         ldp(i,h)    Logical constraints dp
         lvdot(i,h)  Logical constraints on vdot
         lx(i,h)     Logical constraints on x
         lnp(i)    Logical constraints on np
         lns(i)    Logical constraints on ns
         equ_q1(i,h)
         equ_q2(i,h)
         equ_q3(i,h)
         equ_q4(i,h)
         equ_q5(i,h)
         equ_q6(i,h)
```

```
equ_y_dummy_1(k,k2,i)
equ_y_dummy_2(k,k2,i)
equ_y_dummy_3(k,k2,i)
equ_dummyP_1(k,k2,i,h)
equ_dummyP_2(k,k2,i,h)
equ_dummyP_3(k,k2,i,h)
equ_dummyP_4(k,k2,i,h)
equ_dummyvdot_1(k,i,h)
equ_dummyvdot_2(k,i,h)
equ_dummyvdot_3(k,i,h)
equ_dummyvdot_4(k,i,h)
equ_dummydp_1(k,i,h)
equ_dummydp_2(k,i,h)
equ_dummydp_3(k,i,h)
equ_dummydp_4(k,i,h)
lnp_add(i)
lns_add(i)
lz_add;

Objective ..  objvalue
               =e= SUM(h, Prob(h)*SUM(i,
                    C(i)*sum((k,k2), ord(k)*ord(k2)*y_dummy(k,k2,i))
                    + Cd(i)*sum((k,k2), ord(k)*ord(k2)*dummyP(k,k2,i,h))*Pmax(i)
                                      )
                       );

equ_y_dummy_1(k,k2,i)..    y_dummy(k,k2,i) =l= y_ynp(i,k);
equ_y_dummy_2(k,k2,i)..    y_dummy(k,k2,i) =l= y_yns(i,k2);
equ_y_dummy_3(k,k2,i)..    y_dummy(k,k2,i) =g= y_ynp(i,k) + y_yns(i,k2) - 1;
equ_dummyP_1(k,k2,i,h)..    dummyP(k,k2,i,h) =l= y_dummy(k,k2,i);
equ_dummyP_2(k,k2,i,h)..    dummyP(k,k2,i,h) =g= 0;
equ_dummyP_3(k,k2,i,h)..    dummyP(k,k2,i,h) =l= P(i,h);
equ_dummyP_4(k,k2,i,h)..    dummyP(k,k2,i,h) =g= P(i,h) - (1-y_dummy(k,k2,i));
equ_dummyvdot_1(k,i,h)..    dummyvdot(k,i,h) =l= y_ynp(i,k);
equ_dummyvdot_2(k,i,h)..    dummyvdot(k,i,h) =g= 0;
equ_dummyvdot_3(k,i,h)..    dummyvdot(k,i,h) =l= vdot(i,h);
equ_dummyvdot_4(k,i,h)..    dummyvdot(k,i,h) =g= vdot(i,h) - (1-y_ynp(i,k));
equ_dummydp_1(k,i,h)..    dummydp(k,i,h) =l= y_yns(i,k);
equ_dummydp_2(k,i,h)..    dummydp(k,i,h) =g= 0;
equ_dummydp_3(k,i,h)..    dummydp(k,i,h) =l= dp(i,h);
equ_dummydp_4(k,i,h)..    dummydp(k,i,h) =g= dp(i,h) - (1-y_yns(i,k));

equ_q1(i,h).. q1(i,h) =e= sqr(vdot(i,h));
equ_q2(i,h).. q2(i,h) =e= sqr(w(i,h));
equ_q3(i,h).. q3(i,h) =e= power(w(i,h),3);
equ_q4(i,h).. q4(i,h) =e= q2(i,h)*vdot(i,h);
equ_q5(i,h).. q5(i,h) =e= w(i,h)*q1(i,h);
equ_q6(i,h).. q6(i,h) =e= w(i,h)*vdot(i,h);

gP(i,h).. P(i,h)-alpha(i,h)*q3(i,h)-beta(i)*q4(i,h)-gamma(i)*q5(i,h) =e= 0;
gdp(i,h).. dp(i,h)-aa(i,h)*q2(i,h)-bb(i,h)*q6(i,h)-cc(i,h)*q1(i,h) =e= 0;
sumx(h)..      SUM(i,x(i,h)) =e= 1;
gvdot(i,h)..   -x(i,h) + sum(k, ord(k)*dummyvdot(k,i,h)) =e= 0;
gdpc(i,h)..    -y_zz(i) + sum(k, ord(k)*dummydp(k,i,h)) =e= 0;
lw(i,h)     .. w(i,h) - y_zz(i) =l= 0;
lP(i,h)     .. P(i,h)- y_zz(i) =l= 0;
ldp(i,h)    .. dp(i,h) - y_zz(i) =l= 0;
lvdot(i,h) .. vdot(i,h) - y_zz(i) =l= 0;
lx(i,h)     .. x(i,h) - y_zz(i) =l= 0;
lnp(i)   .. sum(k, ord(k)*y_ynp(i,k)) - 3 * y_zz(i) =l= 0;
```

```
lns(i)   .. sum(k, ord(k)*y_yns(i,k)) - 3 * y_zz(i) =l= 0;
lnp_add(i)   .. sum(k, ord(k)*y_ynp(i,k)) - y_zz(i) =g= 0;
lns_add(i)   .. sum(k, ord(k)*y_yns(i,k)) - y_zz(i) =g= 0;
lz_add       .. sum(i, y_zz(i)) =g= 1;



MODEL Pump /ALL/;
Pump.optfile = 1;
Pump.OPTCA = 1e-3;
Pump.OPTCR = 1e-3;
Pump.reslim = 1e9;
Pump.iterlim = 1e5;


OPTION MINLP=BARON;
SOLVE Pump USING MINLP MINIMIZING objvalue;
```

```
*==============================================================================
*============ The GAMS file for the Sarawak gas production problem =============
*==============================================================================


SETS
i "set of sources"
/M4A, M4B, M3A, M3B, M3C, M3D, M3E, M3F, M3G, M3H, M3I, M3J, SEA, SEB/,
j "set of pools"
/M4, M3, SE, M3P/,
k "set of markets"
/T/,
w "supper set containing all properties and the redundant one (8 in total)"
/CO2, N2, H2S, C1, C2, C3, C4, C5+, RED/,
lifeyear "set of system life span years" /1*25/,
* The relationship between h and subh is h=subh^3
h "set of uncertain scenarios" /1*27/,
subh "set of uncertain scenarios for one uncertain parameter" /1*3/;

ALIAS(subh, subh2);
ALIAS(subh, subh3);
ALIAS(j,jj);
ALIAS(j,jjj);
ALIAS(j,j4);
ALIAS(k,k2);


TABLE T_SP(i,j) "topology matrix for from sources to pools"
       M4   M3   SE   M3P
M4A    1    0    0     0
M4B    1    0    0     0
M3A    0    1    0     0
M3B    0    1    0     0
M3C    0    1    0     0
M3D    0    1    0     0
M3E    0    1    0     0
M3F    0    1    0     0
M3G    0    1    0     0
M3H    0    1    0     0
M3I    0    1    0     0
M3J    0    1    0     0
SEA    0    0    1     0
SEB    0    0    1     0
;

TABLE T_SM(i,k) "topology matrix for from sources to markets"
       T
M4A    0
M4B    0
M3A    0
M3B    0
M3C    0
M3D    0
M3E    0
M3F    0
M3G    0
M3H    0
M3I    0
M3J    0
SEA    0
```

```
SEB    0
;

TABLE T_PP(jj,j) "topology matrix for from pools to pools"
         M4   M3   SE   M3P
M4       0    0    0     1
M3       0    0    0     1
SE       0    0    0     1
M3P      0    0    0     0
;

TABLE T_PM(j,k) "topology matrix for from pools to markets"
         T
M4       0
M3       0
SE       0
M3P      1
;

TABLE U_nominal(i,w) "qualities of sources (%)"
        CO2      N2      H2S      C1      C2      C3      C4      C5+   RED
M4A    2.3048  0.2579  4.8e-3  82.2489  7.2965  3.6886  3.1960  1.0025   0
M4B    2.3048  0.2579  4.8e-3  82.2489  7.2965  3.6886  3.1960  1.0025   0
M3A    0.9488  0.4465  3.6e-3  76.2553  7.3721  6.9870  1.1547  6.8320   0
M3B    0.9488  0.4465  3.6e-3  76.2553  7.3721  6.9870  1.1547  6.8320   0
M3C    0.9488  0.4465  3.6e-3  76.2553  7.3721  6.9870  1.1547  6.8320   0
M3D    0.9488  0.4465  3.6e-3  76.2553  7.3721  6.9870  1.1547  6.8320   0
M3E    0.9488  0.4465  3.6e-3  76.2553  7.3721  6.9870  1.1547  6.8320   0
M3F    0.9488  0.4465  3.6e-3  76.2553  7.3721  6.9870  1.1547  6.8320   0
M3G    0.9488  0.4465  3.6e-3  76.2553  7.3721  6.9870  1.1547  6.8320   0
M3H    0.9488  0.4465  3.6e-3  76.2553  7.3721  6.9870  1.1547  6.8320   0
M3I    0.9488  0.4465  3.6e-3  76.2553  7.3721  6.9870  1.1547  6.8320   0
M3J    0.9488  0.4465  3.6e-3  76.2553  7.3721  6.9870  1.1547  6.8320   0
SEA    2.4263  0.1808  0.6e-3  87.6063  3.7230  1.4481  2.3271  2.2879   0
SEB    2.4263  0.1808  0.6e-3  87.6063  3.7230  1.4481  2.3271  2.2879   0
;

TABLE V_up(k,w) "upper bounds on product qualities at the markets (% or mg/m3)"

    CO2     N2    H2S    C1     C2     C3     C4     C5+    RED
T    3     100    100    100    100    100    100    100    100
;

TABLE V_lo(k,w) "lower bounds on the product qualities at the markets (%)"
     CO2     N2     H2S     C1     C2     C3     C4     C5+    RED
T     0      0       0      0      0      0      0       0      0
;


PARAMETERS

T_S(i) "topology vector for sources",

T_P(j) "topology vector for pools",

T_M(k) "topology vector for product terminals",

DiscountRate "rate to calculate the net present value"
/0.12/,
```

```
T_Psplit(j) "indicate if the outlets of the pools split"
/M4    0
 M3    0
 SE    0
 M3P   0/,

D_up_nominal(k) "nominal maximum demands at the final markets (MMscfd)"
/T  850/,

D_up_range(k) "range of the uncertain maximum demands (MMscfd)"
/T  300/,

D_up(k,h) "realizations of maximum demands at the final markets (MMscfd)",

D_lo(k) "minimum demands at the final markets (MMscfd)"
/T  600/,

C_SO(i) "costs of the flows from sources (M$/MMscf)",

C_MO_nominal(k) "nominal prices of the products at markets (1e3$/MMscf)",

C_MO_range(k) "ranges of prices of the products at markets (1e3$/MMscf)",

C_MO(k,h) "prices of the products at markets in different scenarios (1e3$/MMscf)",

Cy_S(i) "cost of source investment (M$)"
/M4A    10
 M4B    10
 M3A    10
 M3B    10
 M3C    10
 M3D    10
 M3E    10
 M3F    10
 M3G    10
 M3H    10
 M3I    10
 M3J    10
 SEA    10
 SEB    10/,

Cy_P(j) "cost of pool investment (M$)"
/M4     10
 M3     10
 SE     10
 M3P    500/,

Cy_M(k)  "cost of market building fee (LNG plant including slugcatcher) (M$)"
/T   10/,

Cy_SP(i,j) "cost of connecting sources and pools (M$)"
/M4A.M4    0
 M4B.M4    0
 M3A.M3    0
 M3B.M3    0
 M3C.M3    0
 M3D.M3    0
 M3E.M3    0
 M3F.M3    0
 M3G.M3    0
```

```
 M3H.M3    0
 M3I.M3    0
 M3J.M3    0
 SEA.SE    0
 SEB.SE    0/,

Cy_SM(i,k) "cost of connecting sources and markets (M$)",

Cy_PP(jj,j) "cost of connecting pools (M$)"
/M4.M3P    16
 M3.M3P    0
 SE.M3P    50/,

Cy_PM(j,k) "cost of connecting pools and markets (M$)"
/M3P.T    40/,

f_SP_up(i,j) "upper bounds on flows from sources to pools (MMscfd)",

f_SP_lo(i,j) "lower bounds on flows from sources to pools (MMscfd)",

f_SM_up(i,k) "upper bounds on flows from sources to markets (MMscfd)",

f_SM_lo(i,k) "lower bounds on flows from sources to markets (MMscfd)",

f_PP_up(jj,j) "upper bounds on flows from pools to pools (MMscfd)",

f_PP_lo(jj,j) "lower bounds on flows from pools to pools (MMscfd)",

f_PM_up(j,k) "upper bounds on flows from pools to markets (MMscfd)",

f_PM_lo(j,k) "lower bounds on flows from pools to markets (MMscfd)"

f_PP_total(jj,j,h) "flows between pools including all the components (MMscfd)",

f_PM_total(j,k,h) "flows from pools to markets including all the components (MMscfd)",

Quality_CO2(k,h) "the CO2 mol% at the markets",

Prob(h) "probability of each scenario",

DT_S(i),

DT_P(j),

DT_M(k),

DT_SP(i,j),

DT_SM(i,k),

DT_PP(jj,j),

DT_PM(j,k),

subProb(subh),

*===== The residue denotes other cost associated with the subsystem =====
*===== (e.g., the downstream facilities to process the products)    =====
residue
/3000/;
```

```
C_MO_nominal(k)=4.480;
C_MO_range(k)=C_MO_nominal(k)*0.2;

f_PP_total(jj,j,h)=0;
f_PM_total(j,k,h)=0;
Quality_CO2(k,h)=0;

f_SP_up(i,j)=1000;
f_SP_lo(i,j)=0;
f_SM_up(i,k)=1000;
f_SM_lo(i,k)=0;
f_PP_up(jj,j)=1000;
f_PP_lo(jj,j)=0;
f_PM_up(j,k)=1000;
f_PM_lo(j,k)=0;

C_SO(i)=0;
Cy_SM(i,k)=0;
T_S(i)=1;
T_P(j)=1;
T_M(k)=1;

DT_S(i)=2*T_S(i);
DT_P(j)=2*T_P(j);
DT_M(k)=2*T_M(k);
DT_SP(i,j)=2*T_SP(i,j);
DT_SM(i,k)=2*T_SM(i,k);
DT_PP(jj,j)=2*T_PP(jj,j);
DT_PM(j,k)=2*T_PM(j,k);
DT_S('M3A')=1;
DT_S('M3B')=1;
DT_S('M3C')=1;
DT_S('M3D')=1;
DT_P('M3')=1;
DT_P('M3P')=1;
DT_M('T')=1;
DT_SP('M3A', 'M3')=1;
DT_SP('M3B', 'M3')=1;
DT_SP('M3C', 'M3')=1;
DT_SP('M3D', 'M3')=1;
DT_PP('M3', 'M3P')=1;
DT_PM('M3P', 'T')=1;
DT_S('SEA')=1;
DT_P('SE')=1;
DT_SP('SEA','SE')=1;
DT_PP('SE','M3P')=1;




* ===== The following are the pressure related parameters =====

Set
wellpara /1*5/;

Table
WPMP(i,wellpara)
        1       2        3       4      5
```

```
M3A   69.33  1.888e-1 3.445e-4 1.482 1.090e3
M3B   82.08  1.662e-1 3.691e-4 1.621 1.215e3
M3C   78.36  1.816e-1 3.481e-4 1.539 1.048e3
M3D   73.13  1.657e-1 3.695e-4 1.53  1.076e3
M3E   79.77  1.627e-1 3.159e-4 1.642 1.209e3
M3F   80.46  1.818e-1 3.815e-4 1.446 1.160e3
M3G   82.30  1.825e-1 3.513e-4 1.663 1.056e3
M3H   76.71  1.749e-1 3.555e-4 1.617 1.193e3
M3I   79.49  1.824e-1 3.375e-4 1.621 1.033e3
M3J   72.10  1.577e-1 3.502e-4 1.673 1.148e3
M4A   75.69  1.724e-1 3.474e-4 1.573 1.143e3
M4B   81.38  1.753e-1 3.316e-4 1.484 1.219e3
SEA   153.40 1.050    4.298e-3 3.515 3.853e2
SEB   141.22 1.154    4.566e-3 3.813 3.845e2
;

Parameters

P_cP_max /153.40/,

rou "The unit conversion coefficient (hm3/day)/(MMscfd)"
/0.0283168/,

omega(j) "The compression model coefficient",
pai_sc   "Pressure at standard conditions (1 atmosphere) (bar)"
/1.013/,
theta_sc "Temperature at standard conditions (K)"
/288.15/
theta_m(j) "Mean operating temperature for compressor j (K)",
eta_j(j) "Compression efficiency for compressor j",
v_nu "Exponential factor for compressor",
tau_sec "Number of seconds in a day (84600 s/d)"
/84600/,
zeta "The polytropic constant the polytropic work of compression"
/1.5/,
CompressPower_up(j) "The upper bound on compressor power (MW)"
/M3P   27/,
CompressPower_lo(j) "The upper bound on compressor power (MW)"
/M3P   0.01/,
CompressMaxPressure(j) "The upper bound on compressor outlet pressure (bar)"
/M3P   200/,

Kappa_PM_nominal(j,k) "The nominal trunkline flow pressure coefficient (bar2.day2/hm6)"
/M3P.T   2.46/,

Kappa_PM_range(j,k) "The range of trunkline flow pressure coefficient (bar2.day2/hm6)"
/M3P.T   0.5/,

Kappa_PM(j,k,h) "Real trunkline flow pressure coefficients (bar2.day2/hm6)"
/M3P.T.1   0/,

DePressure_up(k) "Upper bound on delivery pressure at product terminal (bar)"
/T   80/,
DePressure_lo(k) "Lower bound on delivery pressure at product terminal (bar)"
/T   30/,

gama_HVC(w) "Heating values of components (MJ/kg)"
/CO2 0
 N2  0
 H2S 0
```

```
 C1  55.574
 C2  51.95
 C3  50.37
 C4  49.47
 C5+ 48.72
 RED 0/,

mu_MWC(w) "Molecular weights of components (kg/mole)"
/CO2 0.044010
 N2  0.028020
 H2S 0.034082
 C1  0.016043
 C2  0.030070
 C3  0.044097
 C4  0.058123
 C5+ 0.086177
 RED 0/,

phi "Unit conversion coefficient Mmole/hm3"
/42.2845/,
GasEff_nominal "Nominal efficiency of gas turbine for power generation"
/0.4/,
GasEff_range "Range of efficiency of gas turbine for power generation"
/0.02/,
GasEff(h) "Real range of efficiency of gas turbine for power generation"
/1 0/,

minpressure "Minimum pressure at the inlet of each compressor"
/1/
;

theta_m(j)=315;
eta_j(j)=0.75;
v_nu=(zeta-1)/zeta;
omega(j)=1/eta_j(j)*pai_sc/theta_sc*theta_m(j)/v_nu/tau_sec;









*=========== Generate scenarios for three uncertain parameters ============

D_up(k,h)=D_up_nominal(k);
Kappa_PM(j,k,h)=Kappa_PM_nominal(j,k);
GasEff(h)=GasEff_nominal;
C_MO(k,h)=C_MO_nominal(k);

subProb(subh)$(card(subh)>1 and ord(subh)=1) = errorf(-3+6/card(subh));
subProb(subh)$(card(subh)>1 and ord(subh)>1 and ord(subh)<card(subh))
    = errorf(-3+ord(subh)*6/card(subh)) - errorf(-3+(ord(subh)-1)*6/card(subh));
subProb(subh)$(card(subh)>1 and ord(subh) = card(subh))
    = 1 - errorf(-3+(card(subh)-1)*6/card(subh));

if (card(h)=1,
```

```
        Prob(h)=1;
else
    loop(subh,
      loop(subh2,
        loop(subh3,
               Prob(h)$(ord(h)=ord(subh)+(ord(subh2)-1)*card(subh)
                               +(ord(subh3)-1)*card(subh)*card(subh2))
                    = subProb(subh)*subProb(subh2)*subProb(subh3);

               D_up(k,h)$(ord(h)=ord(subh)+(ord(subh2)-1)*card(subh)
                                 +(ord(subh3)-1)*card(subh)*card(subh2))
                    = D_up_nominal(k)-D_up_range(k)/2+D_up_range(k)/card(subh)/2
                      +(ord(subh)-1)*D_up_range(k)/card(subh);

               Kappa_PM(j,k,h)$(ord(h)=ord(subh)+(ord(subh2)-1)*card(subh)
                                       +(ord(subh3)-1)*card(subh)*card(subh2))
                    = Kappa_PM_nominal(j,k)-Kappa_PM_range(j,k)/2
                      +Kappa_PM_range(j,k)/card(subh2)/2
                      +(ord(subh2)-1)*Kappa_PM_range(j,k)/card(subh2);

               C_MO(k,h)$(ord(h)=ord(subh)+(ord(subh2)-1)*card(subh)
                                 +(ord(subh3)-1)*card(subh)*card(subh2))
                    = C_MO_nominal(k)-C_MO_range(k)/2
                      +C_MO_range(k)/card(subh3)/2
                      +(ord(subh3)-1)*C_MO_range(k)/card(subh3);
                 );
               );
             );
    );


BINARY VARIABLES
y_S(i), y_SP(i,j), y_SM(i,k), y_P(j), y_PP(jj,j), y_PM(j,k), y_M(k);

y_S.fx(i)$(DT_S(i)<2)=DT_S(i);
y_P.fx(j)$(DT_P(j)<2)=DT_P(j);
y_M.fx(k)$(DT_M(k)<2)=DT_M(k);

y_SP.fx(i,j)$(DT_SP(i,j)<2)=DT_SP(i,j);
y_SM.fx(i,k)$(DT_SM(i,k)<2)=DT_SM(i,k);
y_PP.fx(jj,j)$(DT_PP(jj,j)<2)=DT_PP(jj,j);
y_PM.fx(j,k)$(DT_PM(j,k)<2)=DT_PM(j,k);


POSITIVE VARIABLES
f_SP(i,j,h) "flows from sources to pools (MMscfd)",
f_SM(i,k,h) "flows from sources to markets (MMscfd)",
f_PP(jj,j,h) "flows from pools to pools (MMscfd)",
f_PM(j,k,h) "flows from pools to markets (MMscfd)",

P_b(i,h) "Bottom-hole pressure at each well (bar)",
P_t(i,h) "Flowing tubing-head pressure at each well (bar)",
```

```
P_cP(j,h) "Compression inlet pressure for each well platform (bar)",
P_P(j,h) "Pressure at each well platform (after compressor) (bar)",
P_ratio(j,h) "Intermediate varaible in compressor model",
P_M(k,h) "Delivery pressure at each product terminal (bar)",
CompPower(j,h) "The operating power of each compressor (MW)",
PGLoss(j,h) "Gas used for power generation for each compressor (MMscfd)";

VARIABLE
objvalue,
yP(i,h);

f_SP.up(i,j,h)=T_SP(i,j)*f_SP_up(i,j);
f_SM.up(i,k,h)=T_SM(i,k)*f_SM_up(i,k);
f_PP.up(jj,j,h)=T_PP(jj,j)*f_PP_up(jj,j);
f_PM.up(j,k,h)=T_PM(j,k)*f_PM_up(j,k);
P_b.up(i,h)=WPMP(i,'1');
P_t.up(i,h)=WPMP(i,'1');
P_cP.up(j,h)$(ord(j)=4)=WPMP('M3C','1');
P_P.up(j,h)$(ord(j)=4)=CompressMaxPressure(j);
P_ratio.up(j,h)$(ord(j)=4)=exp(v_nu*log(CompressMaxPressure(j)/minpressure));
P_M.up(k,h)=DePressure_up(k);
CompPower.up(j,h)$(ord(j)=4)=CompressPower_up(j);
PGLoss.up(j,h)$(ord(j)=4)=1000;

f_PM.lo(j,k,h)=D_lo('T');
P_t.lo(i,h)=minpressure;
P_b.lo(i,h)=minpressure;
P_cP.lo(j,h)$(ord(j)=4)=minpressure;
P_P.lo(j,h)$(ord(j)=4)=DePressure_lo('T');
P_ratio.lo(j,h)$(ord(j)=4)=1;
P_M.lo(k,h)=DePressure_lo(k);
CompPower.lo(j,h)$(ord(j)=4)=CompressPower_lo(j);






*======= The following are the new variables introduced for relaxation ========

POSITIVE VARIABLES
f_SP_sqr(i,j,h)
P_b_sqr(i,h)
P_t_sqr(i,h)
P_P_sqr(j,h)
P_M_sqr(k,h)
f_PMtotal_sqr(j,k,h)
P_ratio_log(j,h)
fPM_Pratio(j,h)
;

VARIABLES
P_P_log(j,h)
P_cP_log(j,h)
;

f_SP_sqr.up(i,j,h)=sqr(f_SP_up(i,j));
P_b_sqr.up(i,h)=sqr(WPMP(i,'1'));
P_t_sqr.up(i,h)=sqr(WPMP(i,'1'));
```

```
P_P_sqr.up(j,h)$(ord(j)=4)=sqr(CompressMaxPressure(j));
P_M_sqr.up(k,h)=sqr(DePressure_up(k));
P_P_log.up(j,h)$(ord(j)=4)=log(CompressMaxPressure(j));
P_cP_log.up(j,h)$(ord(j)=4)=log(WPMP('M3C','1'));
f_PMtotal_sqr.up(j,k,h)$(T_PM(j,k)=1)=sqr(f_PM_up(j,k));
P_ratio_log.up(j,h)$(ord(j)=4)=v_nu*log(CompressMaxPressure(j)/minpressure);
fPM_Pratio.up(j,h)$(ord(j)=4)=f_PM_up(j,'T')*exp(v_nu*log(CompressMaxPressure(j)/minpressure));

P_b_sqr.lo(i,h)=sqr(minpressure);
P_t_sqr.lo(i,h)=sqr(minpressure);
P_P_sqr.lo(j,h)$(ord(j)=4)=sqr(DePressure_lo('T'));
P_M_sqr.lo(k,h)=sqr(DePressure_lo(k));
P_P_log.lo(j,h)$(ord(j)=4)=log(DePressure_lo('T'));
P_cP_log.lo(j,h)$(ord(j)=4)=log(minpressure);

P_ratio.fx(j,h)$(ord(j) ne 4)=0;
CompPower.fx(j,h)$(ord(j) ne 4)=0;
PGLoss.fx(j,h)$(ord(j) ne 4)=0;
yP.fx(i,h)$(DT_S(i) ne 2)=0;
P_P.fx(j,h)$(ord(j) ne 4)=0;
P_cP.fx(j,h)$(ord(j) ne 4)=0;

P_P_sqr.fx(j,h)$(ord(j) ne 4)=0;
f_PMtotal_sqr.fx(j,k,h)$(T_PM(j,k) ne 1)=0;
P_ratio_log.fx(j,h)$(ord(j) ne 4)=0;
P_P_log.fx(j,h)$(ord(j) ne 4)=0;
P_cP_log.fx(j,h)$(ord(j) ne 4)=0;
fPM_Pratio.fx(j,h)$(ord(j) ne 4)=0;




* =================== Scale the variables ==================

f_SP.scale(i,j,h)=100;
f_SM.scale(i,k,h)=100;
f_PP.scale(jj,j,h)=100;
f_PM.scale(j,k,h)=100;
PGLoss.scale(j,h)=100;
f_SP_sqr.scale(i,j,h)=sqr(f_SP.scale(i,j,h));
f_PMtotal_sqr.scale(j,k,h)=sqr(100);
fPM_Pratio.scale(j,h)=100;

P_b.scale(i,h)=10;
P_t.scale(i,h)=10;
P_cP.scale(j,h)=10;
P_P.scale(j,h)=10;
P_M.scale(k,h)=10;
CompPower.scale(j,h)=10;
P_b_sqr.scale(i,h)=sqr(P_b.scale(i,h));
P_t_sqr.scale(i,h)=sqr(P_t.scale(i,h));
P_P_sqr.scale(j,h)=sqr(P_P.scale(j,h));
P_M_sqr.scale(k,h)=sqr(P_M.scale(k,h));
P_P_log.scale(j,h)=log(P_P.scale(j,h));
P_cP_log.scale(j,h)=log(P_cP.scale(j,h));
yP.scale(i,h)=10;
```

```
EQUATIONS
Objective,

SP_PipeConstr_up(i,j,h),
SM_PipeConstr_up(i,k,h),
Topology_source_SP(i,j),
Topology_source_SM(i,k),
Topology_source_S1(i),
Topology_source_S2(i),

TotalMB_pool_1(j,jjj,h),
TotalMB_pool_3(j,k,h),
CapacityPipe_PP_up(j,jjj,h),
CapacityPipe_PP_up_int(j,jjj,h),
CapacityPipe_PM_up(j,k,h),
CapacityPipe_PM_up_int(j,k,h),
Topology_pool_SP(i,j),
Topology_pool_PP1(jj,j),
Topology_pool_PP2(j,jjj),
Topology_pool_PM(j,k),
Topology_pool_P1_A(j),
Topology_pool_P1_B(j),
Topology_pool_P2_A(j),
Topology_pool_P2_B(j),

Capacity_market_up(k,h),
Capacity_market_lo(k,h)
Topology_market_SM(i,k),
Topology_market_PM(j,k),
Topology_market_M1(k),
Topology_market_M2(k),

Pressure_SP_1(i,j,h),
Pressure_SP_2(i,j,h),
Pressure_SP_3_A(i,j,h),
Pressure_SP_3_B(i,j,h),
yP_definition_1(i,h),
yP_definition_2(i,h),
yP_definition_3(i,h),
yP_definition_4(i,h),
Pressure_P_1(j,h),
Pressure_P_2(j,h),
Pressure_P_3(j,h),
Pressure_PM(j,k,h),
f_SP_sqr_definition(i,j,h)
P_b_sqr_definition(i,h)
P_t_sqr_definition(i,h)
P_P_sqr_definition(j,h)
P_M_sqr_definition(k,h)
f_PMtotal_sqr_definition(j,k,h)
P_ratio_log_definition(j,h)
P_P_log_definition(j,h)
P_cP_log_definition(j,h)
fPM_Pratio_definition(j,h)
```

```
;


Objective..
            objvalue =e= sum(i$(DT_S(i)>0), Cy_S(i)*y_S(i))
                        +sum(j$(DT_P(j)>0), Cy_P(j)*y_P(j))
                        +sum(k$(DT_M(k)>0), Cy_M(k)*y_M(k))
                        +sum((i,j)$(DT_SP(i,j)>0), Cy_SP(i,j)*y_SP(i,j))
                        +sum((i,k)$(DT_SM(i,k)>0), Cy_SM(i,k)*y_SM(i,k))
                        +sum((jj,j)$(DT_PP(jj,j)>0), Cy_PP(jj,j)*y_PP(jj,j))
                        +sum((j,k)$(DT_PM(j,k)>0), Cy_PM(j,k)*y_PM(j,k))
                        +sum(lifeyear, 1e-3*365/power(1+DiscountRate,ord(lifeyear)))*sum(h,
                                Prob(h)*(
                                        sum(i, C_SO(i)*(sum(j$(DT_SP(i,j)>0), f_SP(i,j,h))
                                                        +sum(k$(DT_SM(i,k)>0), f_SM(i,k,h)))
                                           )
                                        -sum(k, C_MO(k,h)*(sum(j$(DT_PM(j,k)>0), f_PM(j,k,h))
                                                        +sum(i$(DT_SM(i,k)>0), f_SM(i,k,h)))
                                            )
                                         )
                                                                                                )
                        +residue;



* ============== Mass balances and topology constraints at sources ==============

SP_PipeConstr_up(i,j,h)$(DT_SP(i,j)=2)..
            y_SP(i,j)*f_SP_up(i,j) =g= f_SP(i,j,h);

SM_PipeConstr_up(i,k,h)$(DT_SM(i,k)=2)..
            y_SM(i,k)*f_SM_up(i,k) =g= f_SM(i,k,h);

Topology_source_SP(i,j)$(DT_S(i)=2 and DT_SP(i,j)=2)..
            y_S(i) =g= y_SP(i,j);

Topology_source_SM(i,k)$(DT_S(i)=2 and DT_SM(i,k)=2)..
            y_S(i) =g= y_SM(i,k);

Topology_source_S1(i)$(DT_S(i)=1 and
      sum(j$(DT_SP(i,j)=1), T_SP(i,j))+sum(k$(DT_SM(i,k)=1), T_SM(i,k))=0)..
            sum(j$(DT_SP(i,j)=2), y_SP(i,j)) + sum(k$(DT_SM(i,k)=2), y_SM(i,k)) =g= 1;

Topology_source_S2(i)$(DT_S(i)=2)..
            sum(j$(DT_SP(i,j)=2), y_SP(i,j)) + sum(k$(DT_SM(i,k)=2), y_SM(i,k)) =g= y_S(i);




* ============== Mass balances and topology constraints at pools ==============

TotalMB_pool_1(j,jjj,h)$(T_PP(j,jjj)=1 and T_Psplit(j)=0)..
            f_PP(j,jjj,h) =e= sum(i$(T_SP(i,j)=1), f_SP(i,j,h))
                              +sum(jj$(T_PP(jj,j)=1), f_PP(jj,j,h));

TotalMB_pool_3(j,k,h)$(T_PM(j,k)=1 and T_Psplit(j)=0)..
            f_PM(j,k,h) =e= sum(i$(T_SP(i,j)=1), f_SP(i,j,h))
                            +sum(jj$(T_PP(jj,j)=1), f_PP(jj,j,h)) - PGLoss(j,h);
```

```
CapacityPipe_PP_up(j,jjj,h)$(T_PP(j,jjj)=1 and DT_PP(j,jjj)<2)..
          f_PP_up(j,jjj) =g= f_PP(j,jjj,h);

CapacityPipe_PP_up_int(j,jjj,h)$(T_PP(j,jjj)=1 and DT_PP(j,jjj)=2)..
          y_PP(j,jjj)*f_PP_up(j,jjj) =g= f_PP(j,jjj,h);

CapacityPipe_PM_up(j,k,h)$(T_PM(j,k)=1 and DT_PM(j,k)<2)..
          f_PM_up(j,k) =g= f_PM(j,k,h);

CapacityPipe_PM_up_int(j,k,h)$(T_PM(j,k)=1 and DT_PM(j,k)=2)..
          y_PM(j,k)*f_PM_up(j,k) =g= f_PM(j,k,h);

Topology_pool_SP(i,j)$(DT_P(j)=2 and DT_SP(i,j)=2)..
          y_P(j) =g= y_SP(i,j);

Topology_pool_PP1(jj,j)$(DT_P(j)=2 and DT_PP(jj,j)=2)..
          y_P(j) =g= y_PP(jj,j);

Topology_pool_PP2(j,jjj)$(DT_P(j)=2 and DT_PP(j,jjj)=2)..
          y_P(j) =g= y_PP(j,jjj);

Topology_pool_PM(j,k)$(DT_P(j)=2 and DT_PM(j,k)=2)..
          y_P(j) =g= y_PM(j,k);

Topology_pool_P1_A(j)$(DT_P(j)=1 and
    sum(i$(DT_SP(i,j)=1), T_SP(i,j))+sum(jj$(DT_PP(jj,j)=1), T_PP(jj,j))=0)..
         sum(i$(DT_SP(i,j)=2), y_SP(i,j)) + sum(jj$(DT_PP(jj,j)=2), y_PP(jj,j)) =g= 1;

Topology_pool_P1_B(j)$(DT_P(j)=1 and
    sum(jjj$(DT_PP(j,jjj)=1), T_PP(j,jjj))+sum(k$(DT_PM(j,k)=1), T_PM(j,k))=0)..
         sum(jjj$(DT_PP(j,jjj)=2), y_PP(j,jjj)) + sum(k$(DT_PM(j,k)=2), y_PM(j,k)) =g= 1;

Topology_pool_P2_A(j)$(DT_P(j)=2)..
         sum(i$(DT_SP(i,j)=2), y_SP(i,j)) + sum(jj$(DT_PP(jj,j)=2), y_PP(jj,j)) =g= y_P(j);

Topology_pool_P2_B(j)$(DT_P(j)=2)..
         sum(jjj$(DT_PP(j,jjj)=2), y_PP(j,jjj)) + sum(k$(DT_PM(j,k)=2), y_PM(j,k)) =g= y_P(j);




* ============= Mass balances and topology constraints at product terminals =============

Capacity_market_up(k,h)$(DT_M(k)=1)..
          D_up(k,h) =g= sum(j$(T_PM(j,k)=1), f_PM(j,k,h))
                       +sum(i$(T_SM(i,k)=1), f_SM(i,k,h));

Capacity_market_lo(k,h)$(DT_M(k)=1)..
          D_lo(k) =l= sum(j$(T_PM(j,k)=1), f_PM(j,k,h))
                       +sum(i$(T_SM(i,k)=1), f_SM(i,k,h));

Topology_market_SM(i,k)$(DT_M(k)=2 and DT_SM(i,k)=2).. y_M(k) =g= y_SM(i,k);

Topology_market_PM(j,k)$(DT_M(k)=2 and DT_PM(j,k)=2).. y_M(k) =g= y_PM(j,k);

Topology_market_M1(k)$(DT_M(k)=1 and
    sum(i$(DT_SM(i,k)=1), T_SM(i,k))+sum(j$(DT_PM(j,k)=1), T_PM(j,k))=0)..
          sum(i$(DT_SM(i,k)=2), y_SM(i,k)) + sum(j$(DT_PM(j,k)=2), y_PM(j,k)) =g= 1;
```

```
Topology_market_M2(k)$(DT_M(k)=2)..
             sum(i$(DT_SM(i,k)=2), y_SM(i,k)) + sum(j$(DT_PM(j,k)=2), y_PM(j,k)) =g= y_M(k);




* ==================== Pressure-related constraints ====================

Pressure_SP_1(i,j,h)$(T_SP(i,j)=1)..
   WPMP(i,'2')*rou*f_SP(i,j,h) + WPMP(i,'3')*sqr(rou)*f_SP_sqr(i,j,h)
         =e= sqr(WPMP(i,'1')) - P_b_sqr(i,h);

Pressure_SP_2(i,j,h)$(T_SP(i,j)=1)..
   WPMP(i,'5')*sqr(rou)*f_SP_sqr(i,j,h)
         =e= P_b_sqr(i,h) - WPMP(i,'4')*P_t_sqr(i,h);

Pressure_SP_3_A(i,j,h)$(DT_S(i) = 1 and ord(j)=4)..
   P_t(i,h) =g= P_cP(j,h);

Pressure_SP_3_B(i,j,h)$(DT_S(i) = 2 and ord(j)=4)..
   P_cP_max + yP(i,h) =g= P_cP(j,h);

yP_definition_1(i,h)$(DT_S(i) = 2)..
   yP(i,h) =l= y_S(i)*(P_t.up(i,h)-P_cP_max);

yP_definition_2(i,h)$(DT_S(i) = 2)..
   yP(i,h) =g= y_S(i)*(P_t.lo(i,h)-P_cP_max);

yP_definition_3(i,h)$(DT_S(i) = 2)..
   yP(i,h) =l= P_t(i,h)-P_cP_max - (1-y_S(i))*(P_t.lo(i,h)-P_cP_max);

yP_definition_4(i,h)$(DT_S(i) = 2)..
   yP(i,h) =g= P_t(i,h)-P_cP_max - (1-y_S(i))*(P_t.up(i,h)-P_cP_max);

Pressure_P_1(j,h)$(ord(j)=4)..
   CompPower(j,h)
         =e= omega(j)*rou*1e5*(fPM_Pratio(j,h)-f_PM(j,'T',h));

Pressure_P_2(j,h)$(ord(j)=4)..
   P_ratio_log(j,h) =e= v_nu*(P_P_log(j,h) - P_cP_log(j,h));

Pressure_P_3(j,h)$(ord(j)=4)..
   CompPower(j,h) =e= sum(w,
     rou*phi/tau_sec*mu_MWC(w)*gama_HVC(w)*1e6*GasEff(h)*PGLoss(j,h)*U_nominal('M3C',w)/100
                         );

Pressure_PM(j,k,h)$(ord(j)=4)..
   P_P_sqr(j,h) - P_M_sqr(k,h)
         =e= Kappa_PM(j,k,h)*sqr(rou)*f_PMtotal_sqr(j,k,h);


f_SP_sqr_definition(i,j,h)$(T_SP(i,j)=1)..
           f_SP_sqr(i,j,h) =e= sqr(f_SP(i,j,h));

P_b_sqr_definition(i,h)..
           P_b_sqr(i,h) =e= sqr(P_b(i,h));

P_t_sqr_definition(i,h)..
           P_t_sqr(i,h) =e= sqr(P_t(i,h));
```

```
P_P_sqr_definition(j,h)$(ord(j)=4)..
          P_P_sqr(j,h) =e= sqr(P_P(j,h));

P_M_sqr_definition(k,h)..
          P_M_sqr(k,h) =e= sqr(P_M(k,h));

f_PMtotal_sqr_definition(j,k,h)$(T_PM(j,k)=1)..
          f_PMtotal_sqr(j,k,h) =e= sqr(f_PM(j,k, h));

P_ratio_log_definition(j,h)$(ord(j)=4)..
          P_ratio_log(j,h) =e= log(P_ratio(j,h));

P_P_log_definition(j,h)$(ord(j)=4)..
          P_P_log(j,h) =e= log(P_P(j,h));

P_cP_log_definition(j,h)$(ord(j)=4)..
          P_cP_log(j,h) =e= log(P_cP(j,h));

fPM_Pratio_definition(j,h)$(ord(j)=4)..
          fPM_Pratio(j,h) =e= f_PM(j,'T',h)*P_ratio(j,h);



MODEL SarawakP
/ALL/;


OPTION MINLP = BARON;
SarawakP.optfile = 1;
SarawakP.iterlim = 1e5;
SarawakP.scaleopt = 1;
SarawakP.optca = 1e-3;
SarawakP.optcr = 1e-3;
SarawakP.reslim = 1e9;


SOLVE SarawakP using minlp minimizing objvalue;
```