TP Metrix - Réseau à 6 noeuds - Enoncé

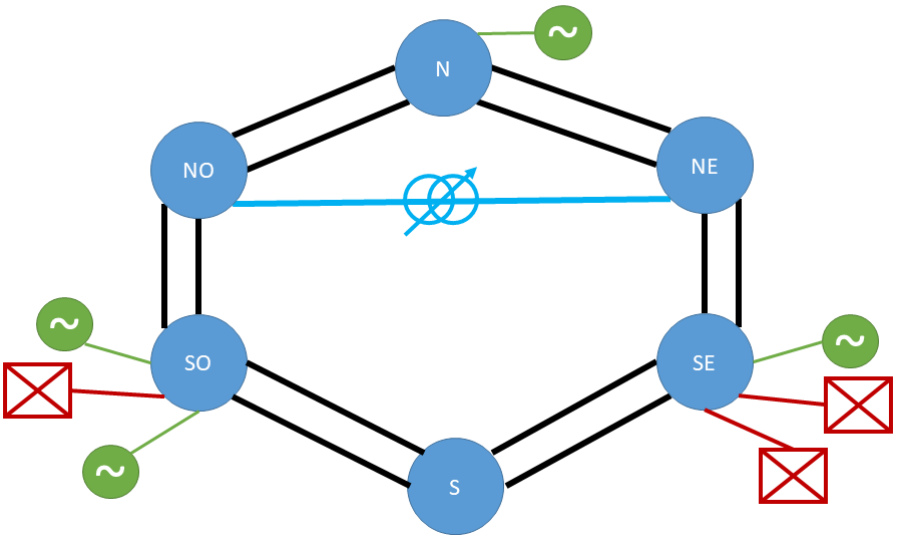
Mathilde Bongrain

November 18, 2020

Objectifs du TP : Sur un réseau très simple (6 postes), s'approprier le paramétrage et les sorties de Metrix, et savoir expliquer les résultats obtenus.

1 Présentation du réseau:

Le réseau utilisé pour ce TP est constitué de 6 postes, tous connectés par deux lignes parallèles de mêmes caractéristiques électrotechniques (même résistances et même réactances pour chaque ligne), ainsi que deux HVDC et un TD. Il comporte aussi 4 groupes et trois charges.



2 Pour bien démarrer:

1. Récupérer les fichiers nécessaires au TP:
   - reseau_6noeuds.xiidm
   - time-series-tp.csv

   Ils sont téléchargeables sur le répertoire de la communauté des études MS: lien communauté ou disponibles sur la clé USB de la formation. Le réseau iidm est celui du schéma précédent et le fichier de chroniques se présente

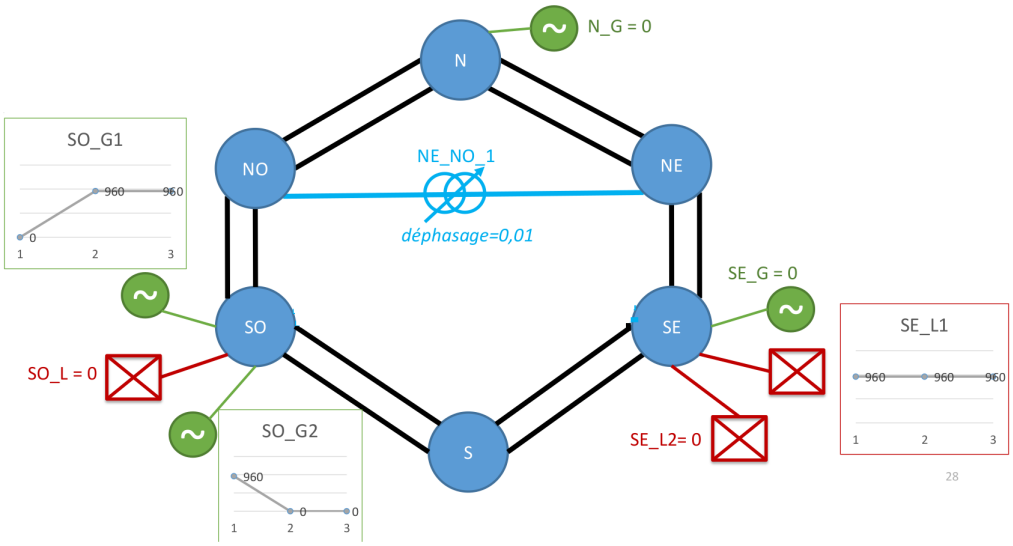
ainsi :

Pas de temps	Version	SE_L1	SO_G1	SO_G2	seuilN	seuilAM
T01	1	960	0	960	400	480
T02	1	960	960	0	400	480
T03	1	960	960	0	100	480

2. Créer un nouveau dossier de travail et recréer l'objet multi-situations :

   - Créer un dossier "TP_Metrix" dans votre projet de travail créé le premier jour de formation
   - (re-)Importer la situation iidm et le fichier de chroniques.
   - Copier le script groovy "Config_MS_reseau-6-noeuds.groovy".
   - Créer un objet "multi-situations" avec ces éléments puis lancer l'analyse du mapping pour vérifier que chaque élément a bien reçu la bonne chronique et que le bilan final est nul.

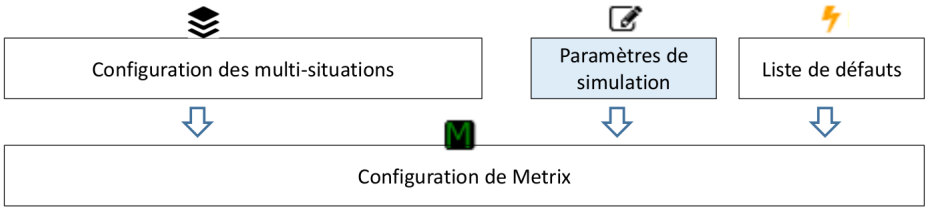
   En sortie de mapping vous devriez avoir le réseau suivant



3 Mode Load Flow :Comprendre les flux

Le load flow Metrix permet de calculer les transits en actif seul sur les ouvrages en N et N-1 sur la base des informations réseau (topologie et caractéristiques électrotechniques), des chroniques de production et consommation et d'une liste de défauts. Il n'optimise rien.

Pour lancer un Load Flow Metrix, il faut donc:



La multi-situation contient les informations réseau et les chroniques.
Le script de configuration metrix est un script qui permet de définir l'ensemble des paramètres et options de simulation (notamment le mode de calcul, les choix de modélisation des ouvrages et les données à afficher en sortie).
Le script de défaut définit la liste de défauts et les options relatifs à ces derniers.

3.1 Action 3A - Lancement en mode LF simple

3.1.1 But:

On souhaite observer les transits sur l'ensemble des ouvrages de notre réseau en N et en N-1 ligne S_SO_1.

3.1.2 En pratique

- Créer un script de configuration Metrix dans lequel on déclare vouloir des résultats sur tous les ouvrages du réseau (si besoin, voir syntaxe ci-dessous).
- Créer une liste de défauts contenant seulement le N-1 sur l'ouvrage S_SO_1 (si besoin, voir syntaxe ci-dessous).
- Créer un calcul Metrix qui pointe vers la multi-situation définie précédemment, le script de configuration et la liste de défauts (voir rappel de la section précédente si besoin).
- Lancer le calcul et noter les transits sur les ouvrages sur les différents pas de temps.

3.1.3 Aide à la syntaxe :

Résultats sur tous les ouvrages réseau:lien wiki
Créer une liste de défauts:lien wiki

3.2 Action 3A - Lancement en mode LF simple - Correction

3.2.1 Scripts:

Script de configuration Metrix:

```
for (l in network.branches) {
   branch(l.id) {
     baseCaseFlowResults true
     maxThreatFlowResults true
```

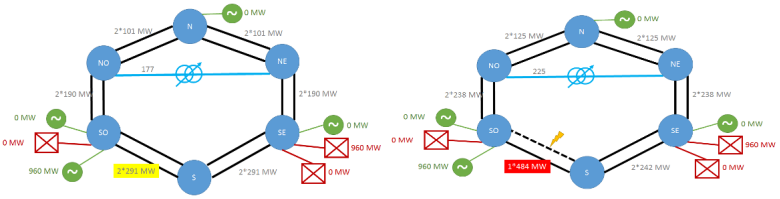
```
    }
}
```

Script de défaut:

```
contingency ('S_SO_1') { equipments 'S_SO_1'}
```

3.2.2 Résultats et Analyse:

Vous devez obtenir les cartes de résulats suivantes:carte des transits en N à gauche et en N-1 à droite.



Plus précisément, sur l'ouvrage S_SO_2 en N et N-1 sur les différents pas de temps, on obtient:

Ts	FLOW_S_SO_2	MAX_THREAT_1_FLOW_S_SO_2	MAX_THREAT_1_NAME_S_SO_2
T01	-290.5	-484.2	S_SO_1
T02	-290.5	-484.2	S_SO_1
T03	-290.5	-484.2	S_SO_1

En N, comme les groupes et les consommations se trouvent au sud, le réseau est moins impédant et on constate que les flux y passent majoritairement. Il y a aussi du flux sur le TD, et au nord. Les trnasits sont les mêmes sur les 3 pas de temps car l'unique différence est que la production passe du groupe SO_G2 au groupe S0_G1 qui sont sur le même poste.

3.3 Action 3B - Surveiller S_SO_2

3.3.1 But:

On souhaite surveiller certains ouvrages (ici S_SO_2), c'est-à-dire observer les transits et les dépassements de seuils uniquement sur l'ouvrage surveillé. Cela permet d'alléger la quantité de résultats à analyser.

3.3.2 En pratique

- Modifier le script de configuration Metrix pour surveiller l'ouvrage S_SO_2 en N et sur incidents.
- Déclarer comme seuil en N et N-k la chronique 'seuilN' fournie dans le jeu de chroniques d'entrées (voir syntaxe).
- Ne plus demander les résultats sur les autres ouvrages pour alléger la lecture des résultats pour la suite.
- Lancer le calcul et analyser les nouveaux résultats.

3.3.3 Aide à la syntaxe :

Déclarer un seuil:lien wiki Description des résultats Metrix : lien wiki

3.4 Action 3B - Surveiller S_SO_2 - Correction

3.4.1 Scripts:

Script de configuration Metrix:

```
branch('S_SO_2') {
   baseCaseFlowResults true // resultats en N
   maxThreatFlowResults true // resultats sur N-k
   branchRatingsBaseCase 'seuilN' // seuil en N
   branchRatingsOnContingency 'seuilN' // seuil en N-k
}
```

3.4.2 Résultats et Analyse:

Vous devez obtenir de nouvelles colonnes dans le fichier de sortie OVERLOAD_BASECASE (qui représentent la différence entre le flux en N et N-1 sur l'ouvrage) et OVERLOAD_OUTAGES (qui représentent la différence entre le flux en N-1 et le seuil).

Ts	OVERLOAD_BASECASE	OVERLOAD_OUTAGES
T01	0	84.2
T02	0	84.2
T03	190.5	384.2

Il n'y a pas de contrainte en N sur les deux premiers pas de temps alors qu'il y en a sur le troisième. C'est dû au changement de la valeur du seuil de 400 à 100 MW dans la chronique 'seuilN'. Il y a des dépassements sur tous les pas de temps en N-1.

4 Mode OPF without redispatching: Optimiser les actions à la main de RTE

4.1 Action 4A - Utiliser les parades topologiques pour résoudre des contraintes

4.1.1 But:

On souhaite voir si les actions à la main de RTE sont suffisantes pour résoudre les contraintes précédemment étudiées. Pour cela, on propose dans un premier d'offrir l'usage des parades topologiques. Dans notre cas, quelles parades topologiques (ouverture de ligne ou passage à deux nœuds dans un poste) pourraient être efficaces ?

4.1.2 En pratique:

- Dans l'objet "simulation Metrix", définir les 4 parades suivantes sur le défaut S_SO_1 (voir syntaxe):

- ouverture du disjoncteur SS1_SS1_DJ_OMN (cela passe à deux nœuds le poste S)
- ouverture du disjoncteur SOO1_SOO1_DJ_OMN (cela passe à deux nœuds le poste de SO)
- ouverture des deux disjoncteurs ci-dessus
- ouverture de la ligne S_SO_2

- Configurer dans le script Metrix le lancement ci-dessous et observer les activations de parades aux différents pas de temps ainsi que l'évolution des contraintes (cf description des résultats Metrix) .

4.1.3 Aide à la syntaxe:

Définition d'une parade sur défaut:lien wiki Description des résultats Metrix: lien wiki

4.2 Action 4A - Utiliser les parades topologiques pour résoudre des contraintes - Correction

4.2.1 Scripts:

Fichiers de parades

```
NB;4;
S_SO_1;1;SS1_SS1_DJ_OMN;
S_SO_1;1;SOO1_SOO1_DJ_OMN;
S_SO_1;2;SS1_SS1_DJ_OMN;SOO1_SOO1_DJ_OMN;
S_SO_1;1;S_SO_2;
```

Fichier de configuration Metrix:

```
parameters {computationType OPF_WITHOUT_REDISPATCHING}

branch('S_SO_2') {
   baseCaseFlowResults true // resultats en N
   maxThreatFlowResults true // resultats sur N-k
   branchRatingsBaseCase 'seuilN' // seuil en N
   branchRatingsOnContingency 'seuilN' // seuil en N-k
}
```

4.2.2 Résultats et Analyse:

Sur le premier pas de temps, Metrix passe à deux nœuds le poste SO, ce qui "allonge le chemin" passant par l'ouvrage contraint et permet de supprimer la contrainte complètement.

Sur les pas de temps 2 et 3, cette parade ne fonctionne pas car la production est sur l'autre nœud du poste SO. Metrix choisit alors la parade qui consiste à ouvrir l'ouvrage en contrainte. Il n'avait pas retenu cette parade sur le premier pas de temps car on l'a déclarée en dessous dans la liste de parades : à efficacité identique, les parades sont choisies dans l'ordre de la liste.

Sur le 3ème pas de temps, la contrainte en N demeure inchangée.

4.3 Action 4B - Utiliser un transformateur déphaseur

4.3.1 But:

Le but ici est de voir si l'usage du transformateur déphaseur permet à lui seul de résoudre les contraintes. Les déphasages des TDs sont par défaut fixés à leur valeur dans la multi-situations (choix de la prise), on peut demander à Metrix de les optimiser en préventif et sur certains incidents. Quel signe de déphasage permettrait de soulager les contraintes N et N-K identifiées ? Positif qui freine le flux entre NO et NE ou négatif qui accentue le flux entre NO et NE

4.3.2 En pratique:

Dans le fichier de configuration Metrix:

- autoriser le TD NE_NO_1 à bouger en préventif et sur le défaut S_SO_1 (voir syntaxe).
- Lancer le calcul sans parades
- Observer les actions sur le TD et l'évolution des contraintes (cf description des résultats Metrix).

Pour la suite du tutoriel, on n'utilisera plus le TD, pensez à le retirer de la configuration Metrix pour les prochaines étapes.

4.3.3 Aide à la syntaxe:

Autorisation d'un TD à bouger en préventif et/ou curatif: lien wiki Description des résultats Metrix : lien wiki

4.4 Action 4B - Utiliser un transformateur déphaseur - Correction

4.4.1 Scripts:

Fichier de configuration Metrix:

```
parameters {
    computationType OPF_WITHOUT_REDISPATCHING
}

branch('S_SO_2') {
   baseCaseFlowResults true // resultats en N
   maxThreatFlowResults true // resultats sur N-k
   branchRatingsBaseCase 'seuilN' // seuil en N
   branchRatingsOnContingency 'seuilN' // seuil en N-k
}

phaseShifter('NE_NO_1') {
  controlType OPTIMIZED_ANGLE_CONTROL
  onContingencies   'S_SO_1'
}
```

4.4.2 Résultats et Analyse:

En lançant le calcul OPF_WITHOUT_REDISPATCHING sans parade, on observe que Metrix joue sur le déphasage du TD en curatif sur tous les pas de temps et en préventif sur le troisième pas de temps. Les contraintes sont entièrement levées :

Résultat	T01	T02	T03
PST_CUR_NE_NO_1_S_SO_1	-0.32	-0.32	-1.01
PST_NE_NO_1	/	/	-0.75
PST_CUR_TAP_NE_NO_1_S_SO_1	1	1	1
PST_TAP_NE_NO_1	/	/	1
OVERLOAD_BASECASE	0	0	0
OVERLOAD_OUTAGES	0	0	0

5 Mode OPF : Optimiser l'ensemble des actions

5.1 Action 5A - Configurer des groupes ajustables en préventif

5.1.1 But:

Le but est ici de s'occuper des contraintes résiduelles après les actions " gratuites" que constituent les parades (on suppose que le TD n'est pas disponible/n'existe pas). On rappelle que dans la partie 4A, les parades n'avaient pas pu résoudre les contraintes en préventif. On propose donc de voir si l'usage des groupes en préventif permettent de résoudre ces contraintes. A quelle puissance des groupes peut-on s'attendre? en préventif ou en curatif?

5.1.2 En pratique:

- Définir dans le fichier de configuration Metrix que tous les groupes peuvent bouger en préventif avec des coûts à la hausse de 100 et à la baisse de 1 (voir syntaxe)
- Changer le mode de simulation en "OPF", reprendre des parades, et lancer le calcul
- Observer les actions prises par Metrix (cf description des résultats Metrix) dont le coût de redispatching

NB : le groupe SE_G a une Pmax de 600MW

5.1.3 Aide à la syntaxe:

Configurer les groupes:lien wiki Description des résultats Metrix : lien wiki

5.2 Action 5A - Configurer des groupes ajustables en préventif - Correction

5.2.1 Scripts:

Fichier de configuration Metrix:

```
parameters {computationType OPF}

branch('S_SO_2') {
   baseCaseFlowResults true // resultats en N
   maxThreatFlowResults true // resultats sur N-k
   branchRatingsBaseCase 'seuilN' // seuil en N
   branchRatingsOnContingency 'seuilN' // seuil en N-k
}

for (g in ['SO_G1','SE_G','SO_G2','N_G']) {
 generator(g) {
   redispatchingDownCosts 1
   redispatchingUpCosts 100
 }
}
```

5.2.2 Résultats et Analyse:

En lançant le calcul OPF avec les parades précédentes et les groupes en préventif, on observe bien que Metrix a utilisé les parades en priorité sur le curatif et permet de résoudre les contraintes en N-k. Pour les écarts en N observés à l'action 4C, il est contraint d'utiliser les groupes en préventif (car les parades ne peuvent pas agir en N). Il y a donc du redispatching sur le 3ème pas de temps; Metrix choisit de baisser le groupe à SO qui débite sur l'ouvrage en contrainte et de monter celui à SE qui est sur le même poste que la charge. Il le monte jusqu'à sa Pmax mais doit ensuite aller chercher le groupe sur le poste N. Le coût de redispatching est égal aux volumes appelés multipliés par leurs coûts.

Au final, il ne reste aucune contrainte (c'est toujours le cas en mode OPF) et le transit sur l'ouvrage surveillé est ramené exactement à 100MW en N sur le 3ème pas de temps.

Résultat	T01	T02	T03
OVERLOAD_BASECASE	0	0	0
OVERLOAD_OUTAGES	0	0	0
GEN_COST	0	0	67695.2
GEN_SO_G1	/	/	-670.2
GEN_SE_G	/	/	600
GEN_N_G	/	/	70.2

5.3 Action 5B - Configurer des groupes ajustables en curatif

5.3.1 But:

Le but est ici de voir comment le redispatching curatif opère en relation avec le redispatching préventif. Afin de voir leur usage, il est nécessaire de supprimer les parades qui ont un coût nul et qui sont donc prioritaires sur n'importe quelle parade coûteuse.

5.3.2 En pratique:

- Configurer des groupes ajustables en curatif
- Préciser que les groupes peuvent également agir sur l'incident 'S_SO_1' (voir syntaxe)
- Supprimer l'usage des parades, et lancer le calcul
- Observer les actions prises par Metrix sur les groupes en préventif et curatif, ainsi que le coût de redispatching

5.3.3 Aide à la syntaxe:

Configurer les groupes:lien wiki

5.4 Action 5B - Configurer des groupes ajustables en curatif - Correction

5.4.1 Scripts:

Fichier de configuration Metrix:

```
parameters {computationType OPF}

branch('S_SO_2') {
   baseCaseFlowResults true // resultats en N
   maxThreatFlowResults true // resultats sur N-k
   branchRatingsBaseCase 'seuilN' // seuil en N
   branchRatingsOnContingency 'seuilN' // seuil en N-k
}

for (g in ['SO_G1','SE_G','SO_G2','N_G']) {
 generator(g) {
   redispatchingDownCosts 1
   redispatchingUpCosts 100
   onContingencies 'S_SO_1'
 }
}
```

5.4.2 Résultats et Analyse:

En gardant les parades, l'ajout des ajustements curatifs ne modifie pas les résultats. En effet, les parades permettent de lever les contraintes à un coût quasi nul comparé aux ajustements de groupes, elles sont donc mises en œuvre de manière prioritaire.

Si on supprime les parades, Metrix réalise également du redispatching curatif sur les deux premiers pas de temps. Sur le 3ème, il ne fait pas d'ajustement curatif car l'ajustement préventif permet déjà d'empêcher les contraintes sur incident.

Résultat	T01	T02	T03
GEN_COST	0	0	67688.1
GEN_SO_G1	/	/	-670.2
GEN_SE_G	/	/	600
GEN_N_G	/	/	70.2
GEN_CUR_SO_G1_S_SO_1	/	-168.5	-266.7
GEN_CUR_SO_G2_S_SO_1	-168.5	/	/
GEN_CUR_SE_G_S_SO_1	168.5	168.5	/
GEN_CUR_N_G_S_SO_1	/	/	266.7

5.5 Action 5C - Retirer un groupe essentiel du curatif

5.5.1 But:

Le but est ici de voir comment Metrix va réagir lorsqu'on retire un groupe essentiel du redispatching en curatif.

5.5.2 En pratique:

- reprendre la même configuration que l'action précédente
- retirer le groupe N_G des groupes ajustables
- reprendre les parades topologiques
- Observer les actions prises par Metrix sur les groupes

5.6 Action 5C - Retirer un groupe essentiel du curatif - Correction

5.6.1 Scripts:

Fichier de configuration Metrix:

```
parameters {computationType OPF}

branch('S_SO_2') {
   baseCaseFlowResults true // resultats en N
   maxThreatFlowResults true // resultats sur N-k
   branchRatingsBaseCase 'seuilN' // seuil en N
   branchRatingsOnContingency 'seuilN' // seuil en N-k
}

for (g in ['SO_G1','SE_G','SO_G2']) {
 generator(g) {
   redispatchingDownCosts 1
   redispatchingUpCosts 100
   onContingencies 'S_SO_1'
 }
}
```

Fichier de parades à prendre en compte.

5.6.2 Résultats et Analyse:

Une fois le groupe de SE_G monté à Pmax, Metrix n'a plus d'autre groupe à disposition pour compenser la baisse du groupe du poste SO. Il coupe alors 35MW de consommation à SE.

Résultat	T01	T02	T03
FLOW_S_S0_2	-290.5	-290.5	-100
LOAD_COST	0	0	386416.7
LOAD_SE_L1	/	/	35.1
GEN_SO_G2	/	/	-635.1
GEN_SE_G	/	/	600
TOPOLOGY_S_SO_1	SOO1_SOO1_DJ_OMN	S_SO_2	S_SO_2

5.7 Action 5D - Autoriser les consommations en préventif

5.7.1 But:

Le but est ici de voir comment Metrix va résoudre les contraintes lorsqu'on autorise du délestage préventif sur SO et seulement les groupes SE (Pmax)600) et SO?

5.7.2 En pratique:

- reprendre la même configuration que l'action précédente (toujours en retirant le groupe N_G des groupes ajustables, et sans parades)
- autoriser les consommations 'SO_L' à bouger en préventif (voir syntaxe)
- Observer les actions prises par Metrix sur les groupes et la consommation

5.7.3 Aide à la syntaxe:

Configurer les consommations:lien wiki

5.8 Action 5D - Autoriser les consommations en préventif - Correction

5.8.1 Scripts:

Fichier de configuration Metrix:

```
parameters {computationType OPF}

branch('S_SO_2') {
   baseCaseFlowResults true // resultats en N
   maxThreatFlowResults true // resultats sur N-k
   branchRatingsBaseCase 'seuilN' // seuil en N
   branchRatingsOnContingency 'seuilN' // seuil en N-k
}

for (g in ['SO_G1','SE_G','SO_G2','N_G']) {
 generator(g) {
   redispatchingDownCosts 1
```

```
13     redispatchingUpCosts 100
14     onContingencies 'S_SO_1'
15   }
16
17   load('SO_L') {
18 preventiveSheddingPercentage 100
19 }
20
21 }
```

Pas de fichier de parades.

5.8.2 Résultats et Analyse:

Dans ce cas, on observe un code erreur 1 sur le troisième pas de temps.
En effet, on a vu juste avant que Metrix devait recourir à du délestage à SE pour trouver une solution, or on n'autorise pas ici ce délestage explicitement mais un autre qui n'est pas utile. Avec les moyens autorisés, Metrix ne trouve aucune solution qui respecte les seuils des ouvrages, il renvoie une erreur.
Cet exemple illustre l'importance de la configuration des moyens d'action. Elle doit être un bon équilibrage entre trop de moyens d'actions (si on en autorise trop, les temps de calculs s'allongent et la compréhension des résultats est difficile) et pas assez (on risque alors de ne pas avoir de solution au problème).

5.9 Action 5E - Autoriser les consommations en curatif

5.9.1 But:

Le but est ici de voir comment Metrix va résoudre les contraintes lorsqu'il se trouve face à plusieurs options. Préférera-t-il l'action de la conso SE_L1 en curatif et préventif ou bien l'action des groupes SO_G1, SO_G1, SE_G et N_G?

5.9.2 En pratique:

- autoriser les groupes SO_G1, SO_G1, SE_G et N_G
- ne pas autoriser les parades
- autoriser la consommation 'SE_L1' à bouger en préventif et curatif (voir syntaxe)
- Observer les actions prises par Metrix sur les groupes et la consommation

5.9.3 Aide à la syntaxe:

Configurer les consommations:lien wiki

5.10 Action 5E - Autoriser les consommations en curatif - Correction

5.10.1 Scripts:

Fichier de configuration Metrix:

```
parameters {computationType OPF}

branch('S_SO_2') {
   baseCaseFlowResults true // resultats en N
   maxThreatFlowResults true // resultats sur N-k
   branchRatingsBaseCase 'seuilN' // seuil en N
   branchRatingsOnContingency 'seuilN' // seuil en N-k
}

for (g in ['SO_G1','SE_G','SO_G2','N_G']) {
 generator(g) {
   redispatchingDownCosts 1
   redispatchingUpCosts 100
   onContingencies 'S_SO_1'
 }

load('SE_L1') {
curativeSheddingPercentage 100
curativeSheddingCost 10
onContingencies 'S_SO_1'
}

}
```

Pas de fichier de parades.

5.10.2 Résultats et Analyse:

Pour rappel, avant cette modification, sans les parades, Metrix devait faire des ajustements de groupes en curatif sur deux premiers pas de temps : il baissait le groupe sur SO et montait le groupe à SE. Les résultats en autorisant la consommation de SE à bouger avec un coût de 10 euros/MW sont :

Résultat	T01	T02	T03
GEN_CUR_SO_G1_S_SO_1	/	-168.5	-133.3
GEN_CUR_SO_G2_S_SO_1	-168.5	/	/
LOAD_CUR_SE_L1_S_SO_1	-168.5	-168/5	-133.3
GEN_SO_G1	/	/	-670.2
GEN_SE_G	/	/	600
GEN_N_G	/	/	70.2

On constate que Metrix préfère alors baisser la consommation à SE plutôt que monter le groupe de SE sur les deux premiers pas de temps en curatif. En effet, on a fixé le coût à la hausse des groupes à 100 euros contre 10 euros pour la baisse de consommation.

5.11 Action 5F - Configurer les seuils avant manoeuvre

5.11.1 But:

Le but est ici de voir comment la définition d'un seuil avant manœuvre peut modifier les actions choisies par Metrix.

5.11.2 En pratique:

- ajouter un seuil de 480 avant curatifsur S_SO_2
- autoriser les groupes SO_G1, SO_G1, SE_G et N_G en préventif et curatif
- ne pas autoriser les parades
- ne pas définir de parade de consommation en préventif ou curatif
- Comparer les actions prises par Metrix par rapport à l'action 5E.

5.11.3 Aide à la syntaxe:

Configurer les sections surveillées:lien wiki

5.12 Action 5F - Configurer les seuils avant manoeuvre - Correction

5.12.1 Scripts:

Fichier de configuration Metrix:

```
parameters {
  computationType OPF
  preCurativeResults true
}

branch('S_SO_2') {
   baseCaseFlowResults true // resultats en N
   maxThreatFlowResults true // resultats sur N-k
   branchRatingsBaseCase 'seuilN' // seuil en N
   branchRatingsOnContingency 'seuilN' // seuil en N-k
   branchRatingsBeforeCurative 480 //seuil avant curatif
}


for (g in ['SO_G1','SE_G','SO_G2','N_G']) {
 generator(g) {
   redispatchingDownCosts 1
   redispatchingUpCosts 100
   onContingencies 'S_SO_1'
 }
}
```

Pas de fichier de parades.

5.12.2 Résultats et Analyse:

Activer le seuil avant manœuvre sans le définir entraine l'apparition d'une nouvelle colonne "MAX_TMP_THREAT_FLOW_S_SO_2" qui contient le pire flux après incident et avant manœuvre. Il est ici de 484 MW sur les deux premiers pas de temps.

Quand on définit un seuil max de 480 MW avant manœuvre, on constate que sur les deux premiers pas de temps, bien qu'il n'y avait ni contrainte en N ni après parade, Metrix fait du redispatching préventif. En effet, le flux avant manœuvre était de 484MW pour un seuil de 480 MW. Metrix effectue donc 8 MW de redispatching préventif pour respecter ce seuil.

Résultat	T01	T02	T03
MAX_TMP_THREAT_FLOW_S_SO_2	-480	-480	-166.65
GEN_COST	852.6	857.2	67695.2
GEN_SO_G1	/	-8.5	-670.2
GEN_SO_G2	-8.4	/	/
GEN_SE_G	8.4	8.4	600
GEN_N_G	/	/	70.2
TOPOLOGY_S_SO_1	SOO1_SOO1_DJ_OMN	S_SO_2	S_SO_2