



La Fédération Française des Jeux Mathématiques

et la

Société de Calcul Mathématique SA

avec l'appui de

Réseau de Transport d'Electricité

Conception d'un réseau de transport d'électricité

Auteurs :

Florian RAMIS
Hugo STRAZIOTA
Sofia ZAOURAR

Etudiants en 2ème année
à Grenoble INP-Ensimag

Encadrant :

Denis NADDEF

Professeur à l'Ensimag
Laboratoire G-SCOP

16 juin 2010

Résumé

Étant donné un ensemble de villes ayant chacune une demande en électricité connue, et un ensemble de centrales de production maximale connue, l'objectif de ce document est de trouver un réseau électrique de coût minimal qui respecte les spécifications du sujet.

Pour résoudre ce problème, nous le modélisons tout d'abord comme un programme linéaire en variables mixtes. Cependant, le nombre exponentiel de contraintes empêche toute résolution par un quelconque solveur. Nous considérons alors dans un premier temps, un problème relaxé où l'on élimine les contraintes "exponentielles". On résout ce problème grâce au logiciel d'optimisation **CPLEX**(IBM-ILOG). Ainsi, les solutions obtenues constituent des bornes inférieures pour notre problème initial sans être forcément réalisables.

Dans un deuxième temps, nous testons la réalisabilité des solutions du problème relaxé pour notre problème initial : cela nous permet de mettre en évidence les contraintes violées. Ces dernières sont ensuite rajoutées au problème relaxé que l'on résout à nouveau... On réitère cette manoeuvre jusqu'à ce que sa solution soit réalisable pour le problème initial. C'est alors la solution optimale recherchée.

Grâce à cette méthode, nous obtenons une solution optimale pour un coût total de l'ordre de **4396 millions d'euros**.

Table des matières

1	Modélisation du problème	3
1.1	Remarques préliminaires	3
1.1.1	Distances euclidiennes	3
1.1.2	Lignes HT et THT	3
1.1.3	Gestion des pannes de transformateurs	4
1.1.4	Position des transformateurs	4
1.1.5	Cas des villes protégées	4
1.1.6	Lignes doubles	5
1.1.7	Points de Steiner	5
1.2	Le modèle	7
1.2.1	Cadre et notations	7
1.2.2	Variables	7
1.2.3	Fonction objectif	7
1.2.4	Contraintes	8
1.2.5	Le programme linéaire en variables mixtes	10
2	Résolution	11
2.1	Algorithme général	11
2.2	Vérification de la réalisabilité de la solution	11
2.2.1	Modélisation du sous-problème	11
2.2.2	Algorithme de vérification	13
2.3	Mise à jour du programme linéaire	13
2.3.1	Panne d'une centrale	14
2.3.2	Panne sur une ligne	14
2.4	Répartition optimale du flot	14
2.5	Récapitulatif des différents modules	15
3	Résultats	16

1 Modélisation du problème

1.1 Remarques préliminaires

Dans cette première partie, nous étudions précisément le cahier des charges, afin de déduire des propriétés de la solution, que nous prendrons en compte dans le modèle.

NB Dans la suite, sauf mention contraire, lorsque nous parlons de villes, nous entendons villes ne se situant pas dans une zone protégée.

1.1.1 Distances euclidiennes

Les distances considérées sont les distances euclidiennes à vol d'oiseau. Elles vérifient en particulier l'inégalité triangulaire. Il est à noter que, lorsque pour relier deux villes la ligne droite passe par une zone protégée, on doit la contourner (pour n'utiliser que des lignes THT). C'est le cas pour la centrale 6 et la ville 6 (**figure 1**).

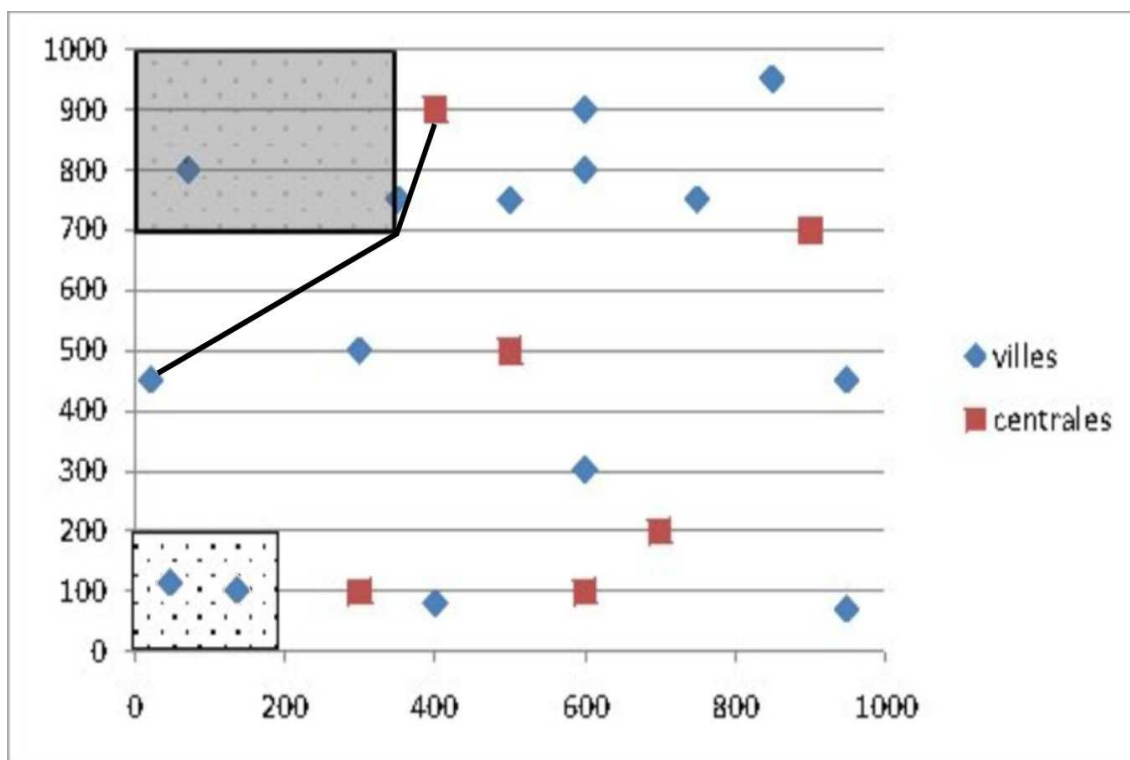


FIG. 1 – Déviation nécessaire pour contourner les zones protégées

1.1.2 Lignes HT et THT

D'après l'énoncé, d'une part, il est obligatoire d'utiliser autant que possible des lignes THT. D'autre part, les villes doivent être alimentées en HT, et les transformateurs doivent être placés à au moins un kilomètre des villes. On en déduit qu'il faut placer les transformateurs à *exactement* un kilomètre des villes et que l'on utilisera seulement des lignes HT de longueur un kilomètre pour relier un transformateur à une ville. Par ailleurs, vu que toutes les villes sont distantes de bien plus d'un kilomètre les unes des autres, nous utiliserons au moins un transformateur par ville.

1.1.3 Gestion des pannes de transformateurs

Si un transformateur ou la ligne HT qui le relie à une ville tombe en panne, cette dernière doit continuer à être alimentée. Nous en déduisons que nous devons prévoir deux transformateurs et deux lignes HT par ville.

1.1.4 Position des transformateurs

Les deux transformateurs associés à une ville sont donc placés sur un cercle de rayon un kilomètre et de centre la ville, qui est considérée ici comme ponctuelle, comme dans le schéma **figure 2**.

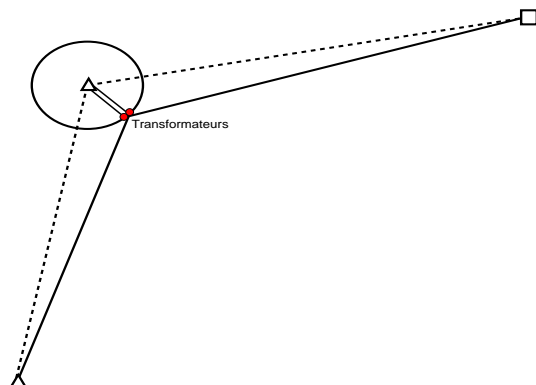


FIG. 2 – *Position des transformateurs*

Or, la plus petite distance entre deux villes, deux centrales ou une ville et une centrale est de l'ordre de 100km et donc bien plus grande qu'un kilomètre. Nous en déduisons que la position exacte (sur le cercle de rayon 1km) des transformateurs n'aura pas d'influence sur les choix à effectuer pour construire le reste du réseau. Ainsi, nous pouvons par exemple considérer que les transformateurs sont superposés aux villes. Puis, on placera les transformateurs à une position optimale une fois le réseau construit.

Par ailleurs, le coût des deux transformateurs et des deux lignes HT associés à une ville est constant. Nous pouvons donc le rajouter au coût du réseau a posteriori.

1.1.5 Cas des villes protégées

Nous savons que dans les zones protégées, nous ne pouvons placer que des lignes HT enterrées. Comme nous utilisons au maximum des lignes THT, nous en déduisons que les transformateurs associés aux zones protégées sont nécessairement sur les frontières.

Nous définissons ainsi un certain nombre de "points d'entrée" dans les zones protégées décrits dans la **figure 3**.

D'autre part, le nombre de transformateurs par zone est au moins de deux. En fait, il y a autant de transformateurs que de lignes HT enterrées arrivant aux villes protégées. Cependant, leur coût, de l'ordre de la moitié d'un kilomètre de ligne THT, est négligeable par rapport au coût des lignes. On peut donc supposer qu'il n'influencera pas l'architecture du réseau. Ainsi, nous nous contenterons de l'ajouter au coût du réseau optimal a posteriori.

A ce stade, nous nous sommes ramenés à un problème ne portant que sur la mise en place des lignes THT.

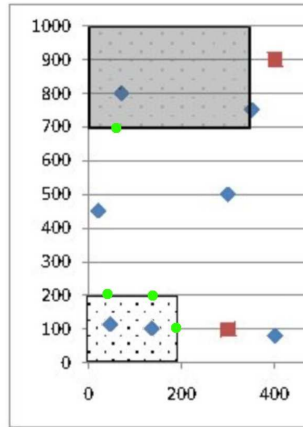


FIG. 3 – Position des points d'entrée aux zones protégées

1.1.6 Lignes doubles

D'abord, vu que les capacités des lignes THT sont plus de trois fois supérieures aux demandes des villes, une seule ligne THT est suffisante pour alimenter une ville à partir d'un autre point du réseau. Néanmoins, pour être résistant au pannes, le réseau peut contenir des lignes doubles entre deux centrales ou entre une centrale et un ville. Par contre, il n'est pas intéressant de permettre la mise en place de lignes doubles entre deux villes. En fait, si une solution contient une ligne double entre deux villes, le réseau est forcément de l'une des deux formes décrites dans les figures 4 et 5.

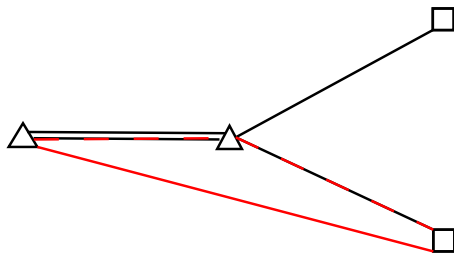


FIG. 4 – Ligne double entre 2 villes - Configuration 1

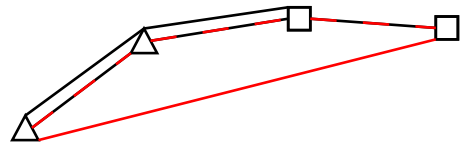


FIG. 5 – Ligne double entre 2 villes - Configuration 2

En effet, d'une part, une ville dépend forcément de deux centrales. D'autre part, les capacités des lignes comparées aux demandes des villes sont telles que nous pouvons toujours alimenter deux villes à partir d'une même ligne.

Au final, dans ces deux configurations, en supprimant les lignes en pointillés et en rajoutant la ligne rouge, vu que les distances sont euclidiennes et que le prix est proportionnel à la distance, nous diminuons le prix du réseau.

1.1.7 Points de Steiner

Nous ajoutons au réseau quelques points où des lignes se séparent, nous les appellerons des points de *Steiner* en référence aux arbres du même nom. On utilise le théorème suivant ¹.

¹cf. article Wikipedia

Théorème Soit ABC un triangle dont les angles sont inférieurs à 120° . Il existe un et un seul point I , tel que la somme $IA + IB + IC$ des distances aux trois sommets soit minimale. Ce point est appelé point de Fermat. Si le triangle contient un angle supérieur à 120° , alors c'est le sommet correspondant qui réalise le minimum.

Positions Compte tenu des contraintes de redondance du réseau, nous faisons l'hypothèse qu'il n'est intéressant de placer des points de Steiner que pour des triangles dont un et un seul des sommets est une centrale. Nous ajoutons au total les neuf points décrits dans la *figure 6* suivante.

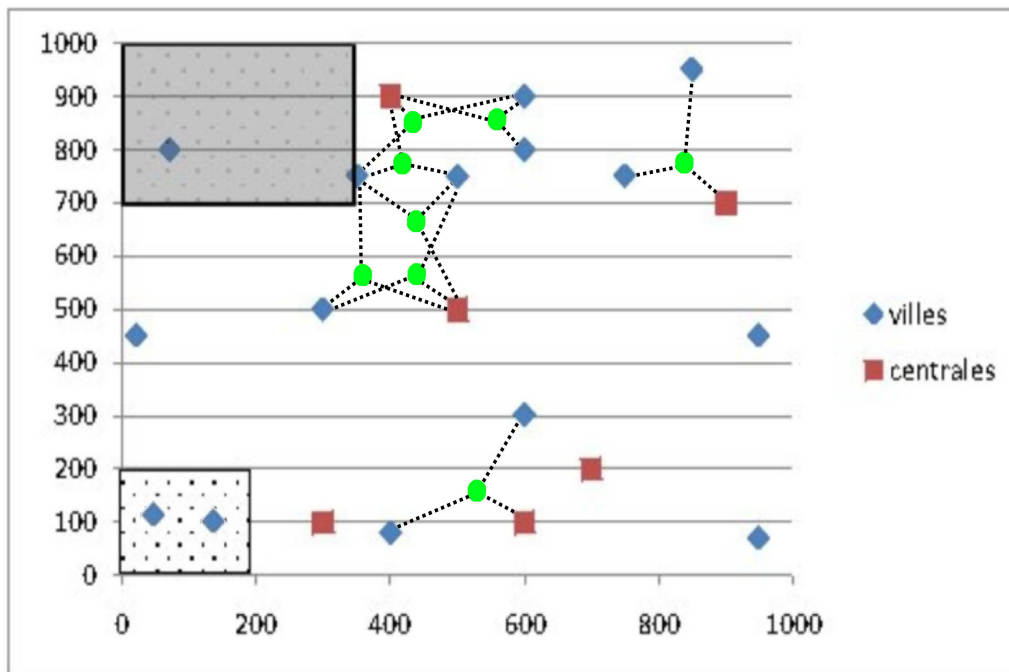


FIG. 6 – Localisation des différents points de Steiner

1.2 Le modèle

1.2.1 Cadre et notations

Soit $G = (X, E)$ un graphe orienté défini de la manière suivante :

- \mathbf{X} l'ensemble des sommets s'écrit : $X = C \cup V \cup VF \cup VP \cup PS$, où :
 - \mathbf{C} représente l'ensemble des centrales,
 - \mathbf{V} l'ensemble des villes en dehors de celles dans les zones protégées,
 - \mathbf{VF} l'ensemble des points d'entrée dans les zones, protégées considérés comme des *villes fictives*,
 - \mathbf{VP} les villes dans les zones protégées,
 - \mathbf{PS} les points de Steiner.
- L'ensemble E contient tous les arcs possibles entre les sommets de C, V, VF ; les arcs entre les sommets de VF et VP dans une même zone; les arcs qui relient un sommet de PS à la centrale et aux 2 villes auxquelles il est associé; les arcs reliant les villes 9 et 10. D'autre part, à chaque sommet, nous associons une "demande algébrique" :

$$\beta_v = \begin{cases} + \text{ production de la centrale } v & \forall v \in C \\ - \text{ demande de la ville } v & \forall v \in V \cup VP \\ 0 & \forall v \in VF \cup PS \end{cases} \quad (1)$$

Et à chaque arc $e \in E$, nous associons une capacité fixe u_e qui correspond à la capacité d'une ligne THT ou HT enterrée.

Le problème revient alors à déterminer un ensemble $E^* \subset E$ qui correspondra aux lignes que nous choisirons de mettre en place dans le réseau.

Remarque. Dire qu'un arc (i, j) est dans E^* revient à ouvrir une ligne physique entre les points i et j . Or, cette ligne peut transporter du courant indifféremment dans un sens ou dans l'autre. Donc, si un arc $(i, j) \in E^*$, alors forcément $(j, i) \in E^*$

1.2.2 Variables

A chaque arc (i, j) , nous associons deux variables :

$$y_{ij} = \begin{cases} 0 & \text{si } (i, j) \notin E^* \\ 1 & \text{si } (i, j) \text{ est un arc simple dans } E^* \\ 2 & \text{si } (i, j) \text{ est un arc double dans } E^* \end{cases}$$

Et :

$$f_{ij} = \text{quantité d'électricité (ou flot) qui circule sur l'arc } (i, j)$$

1.2.3 Fonction objectif

Comme le prix au kilomètre d'une ligne THT ou HT enterrée est le même et vaut 1 million d'euros, en utilisant pour unité le million d'euros, le coût "variables" du réseau s'écrit :

$$\sum_{(i,j) \in E} y_{ij} d_{ij}$$

où d_{ij} représente la distance à vol d'oiseau entre les points i et j .

A celui-ci, il faut ajouter le coût constant dû aux transformateurs ainsi qu'aux lignes HT qui les relient aux villes, comme précisé dans la section 1.1.4. Ce coût comprend pour chaque ville,

le coût de 2 km de lignes HT, et celui de 2 transformateurs. Pour les villes des zones protégées, il faut seulement prendre en compte le coût des transformateurs qui se trouvent sur la frontière de leur zone (cf. section 1.1.5). Pour être précis, il faudrait en déduire, pour chaque ville, le coût de quelques centaines de mètres de ligne THT. On suppose que ce coût est négligeable en effectuant l'approximation suivante.

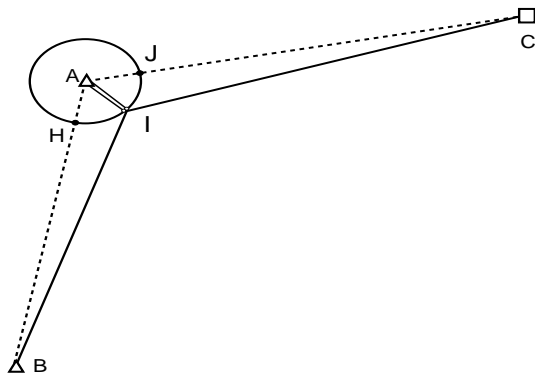


FIG. 7 – Approximation par rapport à la position des transformateurs

Dans la **figure 7**, les lignes THT AH et AJ et la "ligne double" HT AI ont la même longueur qui est exactement 1 km. Par contre, on effectue l'approximation suivante : on considère que la différence de longueur entre les portions BH et BI ainsi que CJ et CI est négligeable par rapport à la longueur totale du réseau.

La fonction objectif, à minimiser s'écrit alors :

$$\sum_{(i,j) \in E, i < j} y_{ij} d_{ij} + cste$$

1.2.4 Contraintes

Le problème comporte plusieurs types de contraintes :

Connexité du réseau

- Une ville doit être reliée à au moins deux lignes pour qu'elle continue à être alimentée en cas de panne de l'une d'elles. Donc :

$$\sum_{i \in V} y_{iv} \geq 2 \quad \forall v \in V \cap VP$$

- Toutes les centrales doivent être utilisées, pour que les demandes soient satisfaites même si la plus grosse centrale tombe en panne. En effet, vu les demandes et les productions, si une centrale, même la plus petite, n'était pas reliée au réseau, alors si la plus grosse centrale tombait en panne, il n'y aurait pas assez d'électricité produite pour satisfaire la demande des villes. D'où la contrainte que chaque centrale doit être reliée au réseau, soit :

$$\sum_{i \in V} y_{iv} \geq 1 \quad \forall v \in C$$

Alimentation des villes Les demandes des villes doivent être satisfaites :

$$\sum_{i \in V} f_{iv} + \beta_v = \sum_{i \in V} f_{vi} \quad \forall v \in V \cup VF \cup VP \cup PS$$

Contraintes de capacité

- Les centrales ne peuvent produire plus que la production spécifiée :

$$\sum_{i \in V} f_{iv} + \beta_v \geq \sum_{i \in V} f_{vi} \quad \forall v \in C$$

- La quantité de courant qui circule sur une ligne doit être inférieure à sa capacité maximale, si la ligne est choisi, égale à 0 sinon. D'où :

$$f_{ij} \leq u_{ij} y_{ij} \quad \forall (i, j) \in E$$

Contraintes de redondance Pour tout sous-ensemble S de X , différentes contraintes sont à ajouter, selon S :

- Si S contient au moins une centrale, il faut que les villes de S continuent à être alimentées en électricité même en cas de panne de la plus "grosse" centrale, que l'on notera c_0 . Donc :

$$\sum_{e \in \delta^-(S)} u_e y_e \geq \sum_{v \in S} (-\beta_v) + \beta_{c_0}$$

2

- Si S contient au moins une ville normale ($v \in V$), il faut que les villes de S continuent à être alimentées en cas de panne d'une quelconque ligne entrant dans S .
Si les capacités des lignes entrant dans S sont toutes égales (à u), cette contrainte s'écrit :

$$\sum_{e \in \delta^-(S)} y_e \geq \left\lceil \frac{\sum_{v \in S} (-\beta_v)}{u} \right\rceil + 1$$

Ce qui revient à :

$$\sum_{e \in \delta^-(S)} y_e \geq \frac{\sum_{v \in S} (-\beta_v)}{u} + 1$$

Et donc :

$$\sum_{e \in \delta^-(S)} u y_e - u \geq \sum_{v \in S} (-\beta_v)$$

Si les capacités sont différentes, il faut raisonner sur les capacités des lignes entrantes nécessaires pour alimenter les villes de S : il faut que le réseau soit tolérant à la panne de chaque ligne ouverte. On explicite cette contrainte lors de la simulation des pannes dans la section 3.3.

On peut donc synthétiser cette contrainte en écrivant :

$$u(\delta^-(S)) \geq \alpha(S) \quad \forall S \subset X$$

Il est à noter qu'il y a autant de contraintes de ce type que de sous ensembles de $C \cup V$. Il y en a donc $2^{|C|+|V|}$, un nombre exponentiel en la taille du problème. Notons ces contraintes C_S .

² $\delta^-(S)$ (resp. $\delta^+(S)$) représente l'ensemble des arcs entrants (resp. sortants) dans l'ensemble de sommets S .

Contraintes liées au modèle

- Une remarque précédente, concernant le rapport entre les arcs du graphes et les lignes physiques du réseau impose la contrainte :

$$y_{ij} = y_{ji} \quad \forall (i, j) \in E$$

- Selon les extrémités de l'arc, y_e peut prendre ses valeurs dans les ensembles $\{0\}$, $\{0, 1\}$ ou $\{0, 1, 2\}$, comme expliqué dans la section 1.1.6.

1.2.5 Le programme linéaire en variables mixtes

Le problème se modélise donc en le programme linéaire en variables mixtes $[\mathcal{P}]$ suivant :

$$\left\{ \begin{array}{lll} & \min & \sum_{(i,j) \in E} d_{ij} y_{ij} \\ \sum_{i \in V} y_{iv} & \geq & 2 \quad \forall v \in V \cap VP \\ \sum_{i \in V} y_{iv} & \geq & 1 \quad \forall v \in C \\ \sum_{i \in V} f_{iv} + \beta_v & \geq & \sum_{i \in V} f_{vi} \quad \forall v \in C \\ \sum_{i \in V} f_{iv} + \beta_v & = & \sum_{i \in V} f_{vi} \quad \forall v \in V \cup VF \cup VP \cup PS \\ f_{ij} & \leq & u_{ij} y_{ij} \quad \forall (i, j) \in E \\ u(\delta^-(S)) & \geq & \alpha(S) \quad \forall S \subset X \\ y_{ij} & = & y_{ji} \quad \forall (i, j) \in E \\ y_{ii} & = & 0 \quad \forall i \in X \\ y_{ij} & \in & \{0, 1, 2\} \quad \forall i \in C, j \in C \cup V, \{(i, j), (j, i)\} \cap E \neq \emptyset \\ y_{ij} & \in & \{0, 1, 2\} \quad \forall i \in VF, j \in VP \text{ (dans une même zone)} \\ y_{ij} & \in & \{0, 1\} \quad \text{pour les autres} \\ f_e & \geq & 0 \quad \forall e \in E \end{array} \right.$$

En pratique, on considère le problème relaxé $[\mathcal{P}']$ où l'on omet les contraintes C_S de redondance du réseau. On résout le problème simplifié, puis on ajoute les contraintes de redondances violées par la solution comme expliqué plus précisément dans la section suivante. S'il y en a pas, vu que la solution du problème simplifié constitue une borne inférieure à la solution du problème complet, c'est une solution optimale.

Il est à noter que dans le problème précédent, un certain nombre de variables sont redondantes et ont nécessité l'ajout de contraintes. Par exemple, seule la partie supérieure (ou inférieure) de la matrice des y_{ij} nous intéresse, vu que cette matrice est symétrique avec une diagonale nulle. En fait, nous avons pu nous le permettre afin de simplifier l'écriture, car ces variables redondantes ainsi que les contraintes associées vont être éliminées lors d'une phase de pré-traitement³ par le logiciel CPLEX utilisé pour le résoudre.

³Plus précisément, nous avons activé le préprocesseur de CPLEX qui contient un "presolver" et un "agregator" qui se chargent d'éliminer les variables et les contraintes inutiles.

2 Résolution

2.1 Algorithme général

La résolution du programme linéaire en variables mixtes relaxé précédent se fait grâce au logiciel CPLEX distribué par IBMTM et aux bibliothèques de **Concert Technology** qui permettent de faire l'interface entre ce logiciel et le langage de programmation C++ que nous avons utilisé.

Notre algorithme de résolution est le suivant :

```
while true do
  (y, f) ← résoudre(problemeLineaire)
  Graphe ← créerGraphe(y,f)
  contraintesViolees ← simulerPannes(Graphe)
  if contraintesViolees = ∅ then
    return SOLUTION OPTIMALE
  else
    problemeLineaire.ajouter(contraintesViolees)
  end if
end while
```

2.2 Vérification de la réalisabilité de la solution

2.2.1 Modélisation du sous-problème

La solution obtenue grâce à CPLEX est solution du programme linéaire $[\mathcal{P}']$ et non de notre programme linéaire de départ $[\mathcal{P}]$. Il nous est donc nécessaire de vérifier si cette solution est réalisable pour notre problème. Ce qui revient à tester si, pour une quelconque panne d'une ligne ou d'une centrale, toutes les villes peuvent tout de même être alimentées. Pour ce faire, nous allons résoudre un problème de circulation dans un graphe. Ce graphe est construit de la manière suivante :

$$G = (V, E) \text{ avec } V = V_1 \cup V_2 \cup V_{f1} \cup V_{f2} \cup V_{p1} \cup V_{p2} \cup \{s, t\}$$

Chaque centrale est associée à un sommet de V_1 , chaque point de Steiner et chaque ville non située dans une zone protégée à un sommet de V_2 , et V_{f1} , V_{p1} (resp. V_{f2} , V_{p2}) correspondent aux villes fictives et protégées de la zone sud (resp. nord). Les sommets s et t sont des sommets artificiels nécessaires pour le problème de circulation.

L'ensemble E des arêtes du graphe est défini ainsi :

$$e = (i, j) \in E \iff \begin{cases} i = s \text{ et } j \in V_1 \\ i, j \in V_1 \times V_1 \\ i \in V_1 \text{ et } j \in V_2 \cup V_{f1} \cup V_{f2} \\ i, j \in (V_2 \cup V_{f1} \cup V_{f2}) \times (V_2 \cup V_{f1} \cup V_{f2}) \\ i \in V_{fk} \text{ et } j \in V_{pk} \quad \forall k \in \{1, 2\} \\ i \in V_2 \cup V_{p1} \cup V_{p2} \text{ et } j = t \\ i = t \text{ et } j = s \end{cases}$$

Les capacités inférieures ou minimales des arcs sont nulles partout sauf pour les arcs partant d'un sommet correspondant à une vraie ville et arrivant sur t pour lesquels la capacité inférieure

est égale à la demande de la ville en MegaWatt. Par conséquent, la capacité inférieure de l'arc de retour (t, s) est égale à la somme des demandes des villes.

Quant aux capacités supérieures ou maximales, elles sont de 1000 pour les arcs représentant des lignes réelles, de 300 pour ceux situés dans les zones protégées. Il n'y a pas de limites pour les arcs entrants et sortants de t , et enfin les capacités des arcs allant de s vers les centrales sont égales à la production de la centrale.

Nous obtenons ainsi le graphe de la **figure 8** :

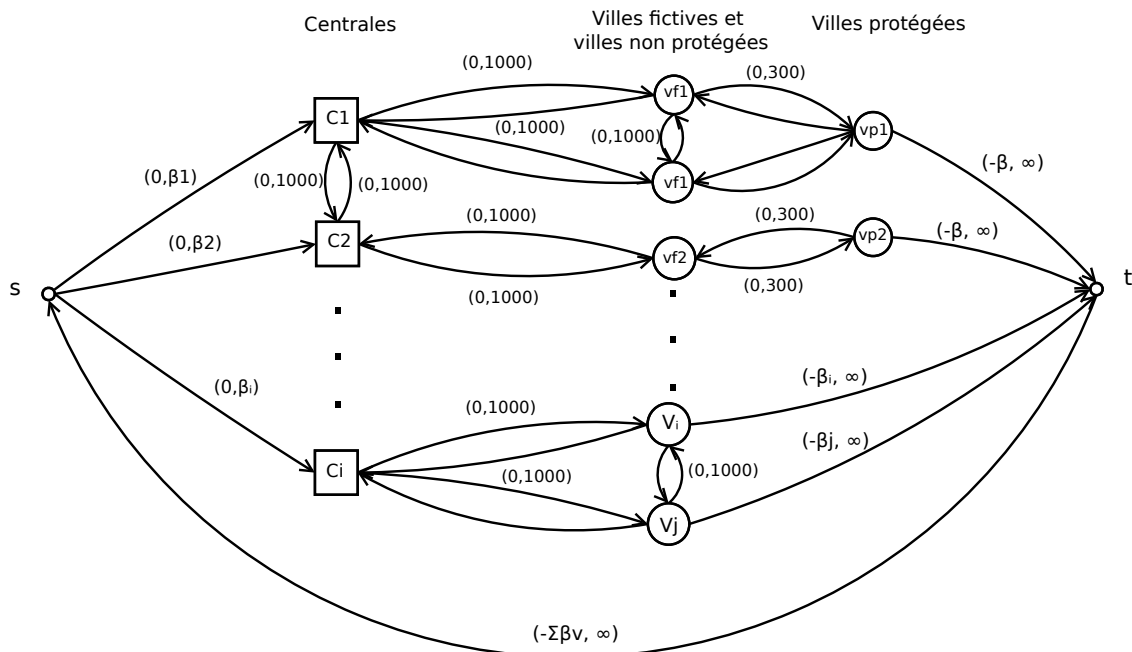


FIG. 8 – Graphe permettant de vérifier si une solution est réalisable

Quatre types de panne sont possibles :

- Une centrale tombe en panne
- Une ligne électrique THT tombe en panne
- Une ligne électrique HT tombe en panne
- Un transformateur tombe en panne

Pour vérifier si la solution obtenue par CPLEX est résistante à ces différentes pannes, nous allons les simuler en modifiant les capacités de certains arcs :

Pour simuler la panne de la centrale i , il suffit de mettre sa production à 0, ce qui revient à mettre la capacité supérieure de l'arc (s, i) à 0.

Les trois autres pannes peuvent être considérées comme une seule et même panne. En effet, comme nous pouvons le voir sur le schéma ci-dessous, si une ligne Haute Tension ou Très Haute Tension tombe en panne, l'électricité ne peut plus circuler dessus, et si un transformateur tombe en panne, on est dans le même cas de figure car un transformateur fait le lien entre la HT vers une ville et la THT vers le reste du réseau.

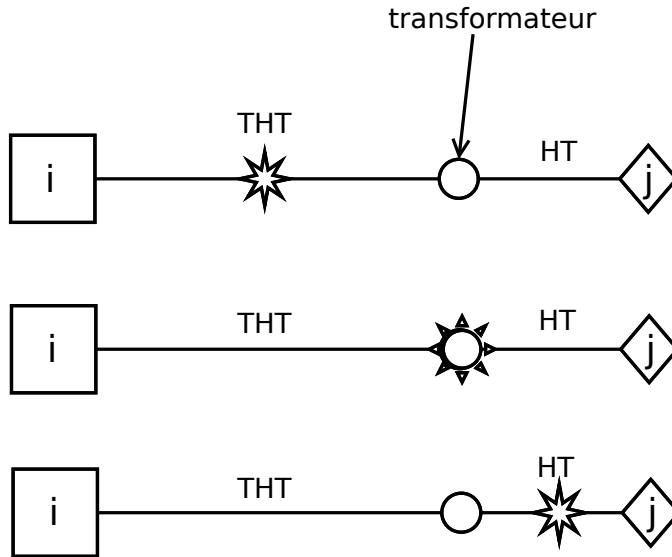


FIG. 9 – Différentes pannes possibles

Dans les deux cas, si nous avons une panne sur la ligne (i, j) nous imposons sa capacité supérieure vaille 0 (et donc qu'il ne circule plus de courant sur cette ligne).

2.2.2 Algorithme de vérification

Une fois la capacité d'une arête (i, j) mise à zéro, nous essayons de retrouver une circulation. Cela revient à essayer de faire passer le flot qui avant circulait sur l'arc (i, j) par un autre chemin allant de i vers j . Pour pousser ce flot de i vers j , nous utilisons l'algorithme de *Ford-Fulkerson* qui consiste en la recherche d'une chaîne augmentante et la mise à jour du flot grâce à cette chaîne. Ici, deux cas sont possibles :

- Soit on réussit à faire passer tout le flot souhaité de i vers j , cela veut dire que la solution est bien résistante à la panne simulée et on passe à la panne suivante.
- Soit l'algorithme ne trouve pas de chaîne augmentante alors que la totalité du flot n'est pas passée. Dans ce cas, l'ensemble des sommets marqués lors de la dernière itération viole la condition d'existence d'une circulation du théorème de circulation d'Hoffman, ce qui nous permet d'ajouter une contrainte au programme linéaire $[\mathcal{P}']$.

Afin d'optimiser le temps d'exécution de notre programme, nous testons toutes les pannes possibles et ajoutons toutes les contraintes correspondantes avant de relancer la résolution par CPLEX.

2.3 Mise à jour du programme linéaire

Le théorème de circulation d'Hoffman nous dit la propriété suivante :

Etant donné un graphe orienté $G = (V, E)$, tel que chaque arc e admet une capacité minimale l_e , une capacité maximale u_e .

Il existe une circulation (ie. une fonction f associant à chaque arête un flot qui satisfait la loi de Kirshoff de conservation en chaque noeud et telle que $\forall e \in E, l_e \leq f_e \leq u_e$) et qui vérifie si et seulement si

$$\forall S \subsetneq V, \sum_{e \in \delta^-(S)} l_e \leq \sum_{e \in \delta^+(S)} u_e$$

Lorsque notre algorithme de simulation des pannes ne trouve pas de circulation, nous obtenons un ensemble de sommets marqués qui ne respectent pas cette condition. A partir de cet ensemble et suivant le type de la panne, nous en déduisons la contrainte à rajouter au programme linéaire.

2.3.1 Panne d'une centrale

Si la centrale i_0 tombe en panne et que l'ensemble de sommets obtenu est S , nous rajoutons au programme linéaire la contrainte suivante :

$$\sum_{i \notin S} \sum_{j \in S} y_{ij} u_{ij} \geq \sum_{v \in S} (-\beta_v) + \beta_{i_0}$$

où β_v correspond soit à la production de la centrale v si $v \in V_1$, soit à l'opposé de la demande de la ville v si $v \in V_2 \cup V_{p1} \cup V_{p2}$, et vaut zéro sinon.

Cela correspond bien à la condition du théorème d'Hoffman, car dans notre graphe seules les capacités inférieures des arcs allant des villes vers t ne sont pas nulles, mais valent la demande des villes. Et les capacités supérieures sont les u_{ij} ainsi que les productions des centrales pour les arcs sortants de s que l'on retrouve dans la partie droite de l'inéquation avec le signe $-$.

2.3.2 Panne sur une ligne

Concernant une panne sur une ligne (i_0, j_0) , et l'ensemble de sommets associé S , la contrainte est très analogue à la précédente :

$$\sum_{i \notin S} \sum_{j \in S} y_{ij} u_{ij} - u_{i_0 j_0} \geq \sum_{v \in S} (-\beta_v)$$

Ce qui correspond bien à la condition du théorème d'Hoffman en prenant en compte les mêmes remarques que pour la panne d'une centrale.

2.4 Répartition optimale du flot

Nous avons rajouté à la solution optimale une distribution de flot optimale : les pertes par effet de Joule sont inférieures lorsque le courant parcourt une distance plus faible. Nous avons alors rajouté un programme linéaire qui, une fois le réseau optimal trouvé, calcule la distribution optimale dans ce réseau :

$$\left\{ \begin{array}{ll} \min \sum_{e \in E} l_e f_e & \\ \sum_{i \in V} f_{iv} + \beta_v \geq \sum_{i \in V} f_{vi} & \forall v \in C \\ \sum_{i \in V} f_{iv} + \beta_v = \sum_{i \in V} f_{vi} & \forall v \in V \cup VF \cup VP \cup PS \\ f_e \leq y_e * u_e & \forall e \in E \end{array} \right.$$

y_e = nombre de lignes sur l'arête e

l_e = longueur de la ligne e

2.5 Récapitulatif des différents modules

En pratique, nous avons implémenté plusieurs classes qui gèrent les différentes étapes décrites précédemment. La *figure10* représente leur organisation.

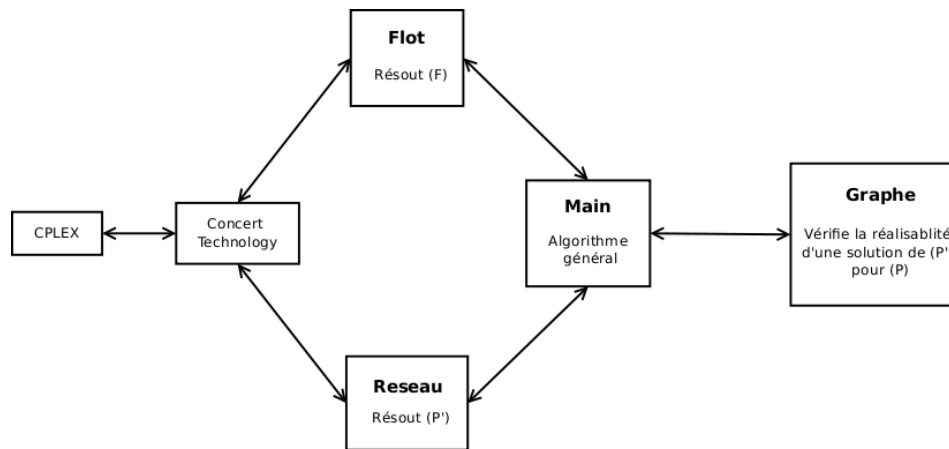


FIG. 10 – Architecture générale de l'application

3 Résultats

Après exécution de notre programme et interprétation nous obtenons le réseau ci-dessous pour un coût total de **4396 millions d'euros**.

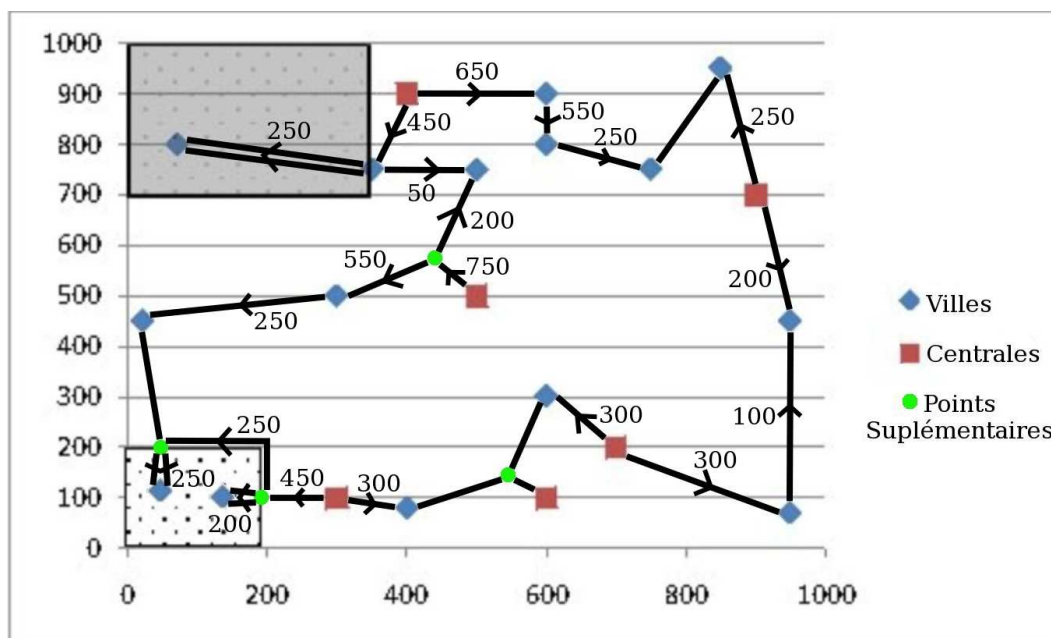


FIG. 11 – Réseau obtenu après exécution du programme principal

Nous constatons que la solution obtenue utilise des points de Steiner pour minimiser les distances, donc le rajout de ces points était bien judicieux pour notre problème.

Nous avons synthétisé les principales données significatives et intéressantes de l'exécution de notre programme :

Temps d'exécution	10 heures
Nombre d'itérations	123
Nombre moyen de contraintes ajoutées par itération	21
Nombre total de contraintes ajoutées	2755

Le logiciel a été exécuté sur la machine Ensibull (serveur BULL Novascale 5160) de l'Ensimag qui a les caractéristiques suivantes :

- 8 processeurs titanium 2 1,5 GHZ, 3 MO cache L3, cache L2 256 Ko
- 32 Go mémoire SDRAM DDR 266
- baie disque FDA 1300 (connexion Fiber Channel) : 10 disques 73 Go 15 ktpm

La répartition du flot optimale trouvée par la résolution du deuxième programme linéaire a été rajoutée au schéma. Ainsi les pertes dues à l'effet Joule sont également minimales. De plus, les coordonnées des transformateurs et des points de Steiner, sont indiquées en annexe.

Conclusion

Dans ce projet nous avons commencé par modéliser le problème industriel proposé. Pour le résoudre, nous avons d'une part utilisé le logiciel d'optimisation le plus répandu dans le domaine de la recherche opérationnelle, **CPLEX**, et d'autre part implémenté un algorithme théorique vu en cours, l'algorithme de recherche d'une circulation dans un graphe. Au final, nous obtenons une solution qui nous semble satisfaisante.

Cependant, il est à noter que, certaines configurations possibles ne sont pas prises en compte dans notre modèle. Par exemple, des points de Steiner de degré supérieur à trois ou dans des triangles où tous les sommets sont des villes.

En fait, nous avons dû trouver un équilibre entre complexité, précision du modèle et temps d'exécution. En effet, nous étions conscient que l'algorithme utilisé par **CPLEX**, bien qu'optimisé au maximum, n'est autre qu'un algorithme de Branch & Cut dont la complexité est exponentielle. Une conséquence par exemple est que, lorsque l'on est parti d'un modèle basique (sans points de Steiner) puis que l'on a rajouté une dizaine de points supplémentaires, le temps d'exécution du programme est passée de quelques minutes à une dizaine d'heures. Ainsi, notre méthode de résolution est limitée à des instances de petites taille.

D'autre part, ce projet était particulier dans le sens où l'on devait trouver une solution optimale pour une unique instance. Nous avons tout de même tenté de suivre une démarche assez indépendante de l'instance pour obtenir une méthode de résolution générique. En fait, nos raisonnements et programmes ont été les "plus génériques possibles", sauf lorsque utiliser des propriétés de l'instance permettait d'améliorer les performances.

Bibliographie

- V. Chvatal *Linear Programming*, 1983.
- R.K Ahuja T.L Magnanti J.B. Orlin, *Network Flows : Theory, Algorithms, and Applications*, 1989.
- Article Wikipedia sur les Points de Fermat http://en.wikipedia.org/wiki/Fermat_points

Annexe

Coordonnées des points supplémentaires	x	y
Transformateurs de V1	200	100
Transformateurs de V2	400	81
Transformateurs de V3	949	71
Transformateurs de V4	28	200
Transformateurs de V5	600	299
Transformateurs de V6	21	449
Transformateurs de V7	300	501
Transformateurs de V8	949	450
Transformateurs de V10	351	751
Transformateurs de V11	499	749
Transformateurs de V12	601	801
Transformateurs de V13	599	899
Transformateurs de V14	750	751
Transformateurs de V15	850	949
Point de Steiner Nord	452	546
Point de Steiner Sud	564	133