Smart Grid : algorithmes

Optimisation des Smart Grids

G. Guérard[1]

[1]PRiSM
UVSQ

9 octobre 2012

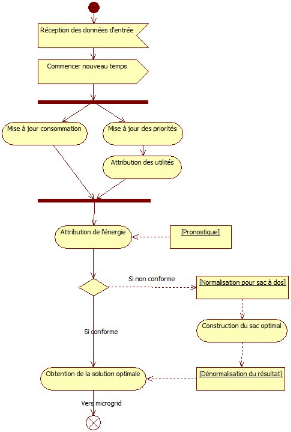
Sommaire

Lignes directrices

Niveau local

- Gestion de la consommation des bâtiments possèdant une interface Smart Meter
- Répartition de l'énergie aux appareils

Organisation du problème



Organisation des appareils

1. ρ_e la priorité de l'appareil :
   1. 0 appareil en marche,
   2. < 1 appareil en besoin énergétique,
   3. > 1 pas de besoin immédiat.
2. $conso_e$ la consommation de l'appareil, cette consommation peut être variable.
3. α_e le coefficient de priorité : détermine la variation de ρ_e sur le temps.

Problème de sac à dos

Données : $2n$ entiers $\{u_i | u_i > 0, i = 1, \ldots, n\}\{p_i | p_i > 0, i = 1, \ldots, n\}$ et un entier $C > 0$.

Question : Trouver un sous-ensemble X de $\{1, \ldots, n\}$ tel que $\sum_{x \in X} p_x \leq C$ et qui maximise $\sum_{x \in X} u_x$.

1. Utilisation d'un vecteur binaire :
   1. si $x_i = 0$ on ne prend pas l'objet,
   2. si $x_i = 1$ on prend l'objet.
2. Utilisation d'une méthode de résolution par programmation dynamique.

Lignes directrices

1. Niveau local
   - Attentes et besoins
   - Résolution par KP
   - Exemple
2. Niveau microgrid
   - Attentes et besoins
   - Pseudo-enchères
   - Equilibre du système
3. Niveau T&D
   - Attentes et besoins
   - Module d'ajustement
   - Equilibre de l'offre et la demande

Niveau local et sac à dos

1. Calcul de l'utilité par nivelation du coût et normalisation des poids.
2. Numération des appareils de 1 à n.
3. Capacité du sac = énergie prévu normalisé.
4. Pas de KP local si les pronostiques sont justes.

Programmation dynamique

Problème

On pose $F_k(E)$ le meilleur coût pour remplir un sac de taille E avec les k premiers objets. La récurrence montre que $F_k(E) = max\{F_{i=k-1}(E), F_{k-1}(E-p_i)+u_i\}$, en effet l'ajout d'un nouvel objet augmente ou non le meilleur coût. Nous cherchons à calculer $F_n(C)$.

Programmation dynamique

Pour $E = 0$ à C Faire
  Si $p[n] \leq E$ alors
    $F[n][E] = p[n]$
  Sinon
    $F[n][E] = 0$
Pour $k = n - 1$ à 1 faire
  Pour $E = 0$ à C faire
    $F[k][E] = F[k+1][E]$
    Si $p[k] \leq E$ et $u[k] + F[k+1][E - p[k]] > F[k+1][E]$ alors
      $F[k][E] = w[k] + F[k+1][E - p[k]]$
    Fin Si
  Fin Pour
Fin Pour
Retourner $F[1][C]$

Obtention du sac optimal

$E = C$
Pour $k = 1$ à $n-1$ faire
  Si $F[k][E] = F[k+1][E]$ alors
    $x[k] = 0$
  Sinon
    $x[k] = 1,\ E = E - p[k]$
Si $F[n][E] = 0$ alors
  $x[n] = 0$
Sinon
  $x[n] = 1$
Retourner x

Avantages et Inconvénients

1. Avantages
    1. Résolution par une méthode exacte : obtention de l'optimum global
    2. Répartition de la ressource en fonction de l'utilité et du poids des objets
    3. Calcul rapide pour de petite instance : de poids entiers et de capacité modérée
2. Inconvénients
    1. Gourmand en mémoire pour des problèmes de grande taille
    2. Résultat optimal au niveau local, mais pas au niveau global : flexibilité en faveur des enchères

Lignes directrices

1. Niveau local
   - Attentes et besoins
   - Résolution par KP
   - Exemple
2. Niveau microgrid
   - Attentes et besoins
   - Pseudo-enchères
   - Equilibre du système
3. Niveau T&D
   - Attentes et besoins
   - Module d'ajustement
   - Equilibre de l'offre et la demande

Exemple

$C = 8$, nous disposons des éléments suivants :

i	1	2	3	4	5
p_i	3	4	5	4	2
u_i	15	18	20	12	6

Construction du sac

Le premier algorithme fabrique le tableau suivant :

E|i	1	2	3	4	5
0	0	0	0	0	0
1	0	0	0	0	0
2	6	6	6	6	6
3	15	6	6	6	6
4	15	18	12	12	6
5	21	20	20	12	6
6	24	24	20	18	6
7	33	26	26	18	6
8	35	30	26	18	6

Appareils utilisés

La solution optimale de 25 est obtenu avec les éléments 1 et 3 :

k	1	2	3	4	5
E	8	5	5	0	0
x[k]	1	0	1	0	0

La $conso_{moy}$ est de 5+3=8.

Lignes directrices

Niveau microgrid

- Gestion du marché entre les niveaux locaux sous la direction du microgrid et les producteurs.
- Equilibre entre l'offre et la demande, flexibilité.

Gestion des niveaux locaux

- Le graphe est construit comme un réseau de jeu :
  - Chaque niveau local appartient à un microgrid.
  - Chaque microgrid possède plusieurs niveaux locaux.
- Le microgrid a accès aux données des niveaux locaux en relation avec lui (consommation de l'instant, vecteur de consommation des appareils).
- Recherche d'un équilibre entre l'énergie fournie et l'énergie consommée.

Pseudo-enchères

1. Chaque joueur envoit son vecteur de consommation au microgrid directement lié.
2. Chaque microgrid stocke les vecteurs de consommation de ses joueurs (dans un tableau).
3. Les microgrids effectuent une enchère pour chacun des joueurs.
4. Les microgrids (sous-stations) renvoient l'enchère final au niveau du réseau T&D (station). Une sous-station est relié qu'à une seule station.
5. Les stations additionnent les enchères des sous-stations desservies. Ce résultat forme la consommation de ce « puit » dans la gestion du niveau T&D.

Lignes directrices

Flexibilité et loi normale

Définition

Une variable aléatoire réelle X suit une loi normale d'espérance μ et d'écart type σ strictement positif si cette variable aléatoire réelle X admet pour densité de probabilité la fonction $p(x)$ définie, pour tout nombre réel x, par :
$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$. On note alors $X \sim \mathcal{N}(\mu, \sigma^2)$. Une telle variable aléatoire est alors dite variable gaussienne.

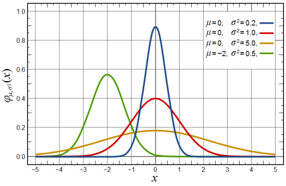


Figure : Densité de probabilité de la loi normale.

Flexibilité et loi normale

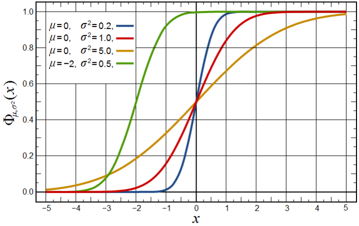


Figure : Fonction de répartition de loi normale.

1. $\mu = U_{moy_i}$
2. 99,7 % de la population est dans l'intervalle $[\mu - 3\sigma, \mu + 3\sigma]$. Soit une marge de ε (pourcentage de $conso_{moy_i}$) autour de $conso_{moy_i}$, 99,7% de la population doit se trouver dans cette marge, donc $3\sigma = \varepsilon$.

Enchères

$conso = 0$

Pour chaque i niveau locale

$\Phi(x*) = aléa[0 \ldots 1]$

$X* = approximation\,\Phi(x*)$

$X_i = X^* * \frac{\varepsilon}{3} + conso_{moy_i}$

Si $X_i < conso_{min_i}$ alors $X_i = conso_{min_i}$

Si $X_i > conso_{max_i}$ alors $X_i = conso_{max_i}$

$conso = conso + X_i$

stocker X_i

Stratégie de consommation

Prenons par exemple un niveau locale ayant pour loi normale $\mathcal{N}(200, 400)$. Il tire un nombre aléatoire de $0,31$. D'après l'approximation de la fonction de répartition de la loi normale réduite, nous en déduisons que $X* = -0,5$. Sa demande de consommation réelle est donc $X = -0,5 * 20 + 200 = 190$ alors que son $conso_{moy}$ est de 200.

1. Stratégie de garder X_i
2. Stratégie de la plus proche consommation brute
3. Stratégie de la plus proche consommation avec coût minimum
4. Etc.

Récompenses et punitions

1. Les consommations sont sommées à chaque microgrid, et forment la consommation du microgrid.
2. Le microgrid garde en mémoire les enchères de chaque niveau local pour effectuer les changements par rétroaction.
   1. Si le niveau local est en sur-consommation, on modifie la loi normale des sur-consommateurs $\mathcal{N}(\mu-\varepsilon,\sigma^2)$,
   2. Si le niveau local est en sous-consommation, on modifie la loi normale des sous-consommateurs $\mathcal{N}(\mu+\varepsilon,\sigma^2)$.
3. Une nouvelle phase d'enchère est lancée, on retourne au point 1.

Lignes directrices

Décision finale

1. Chaque niveau local effectue un problème de sac à dos
2. Le microgrid effectue un problème de sac à dos parmis tout ses niveaux locaux avec l'énergie restante et les appareils non fournis

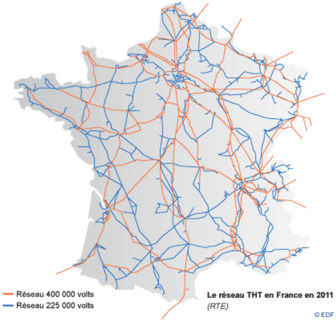
$conso_{restant} = 0$
Pour chaque niveaux locales du microgrid
  Faire knapsack avec $C = conso'_{moy_L}$
  $conso_{restant} = conso_{restant} + (conso' - F_L[1][conso'_{moy_L}])$
  Enlever les appareils consommateurs
Faire knapsack avec $C = conso_{restant}$ avec les appareils restant
Construction du sac optimal : retourner x au niveau local

Avantages et Inconvénients

1. Avantages
   1. Résolution par enchères successives : flexibilité et atteinte d'un équilibre global
   2. Répartition de la ressource en fonction des besoins de chaque utilisateur
   3. Calcul rapide pour de petite instance
2. Inconvénients
   1. Gourmand en mémoire pour des problèmes de grande taille
   2. Surplus énergétique inutilisable : flexibilité localisée à l'instant suivant

Lignes directrices

Attentes et besoins



- Satisfaire l'offre et la demande, gestion du marché de l'énergie.
- Eviter la congestion, la surconsommation ou la sousconsommation (production).

Construction du graphe

1. Les producteurs sont des Sources.
2. Les microgrids sont des Puits.
3. Les lignes ont des capacités de transit.

Problème

Résolution d'un flot max dans un graphe. Il faut alors modifier notre graphe pour correspondre à un problème de Ford-Fulkerson

Problème de flot max

Soit $G = (V, A)$ un graphe orienté. Soit s la source et t le puit de G. On associe à chaque arête (u, v) de G une capacité c_{uv} qui represente le flot maximum pouvant passer par cette arête. Cette capacité est positive et on note c le vecteur dans $\mathbb{R}_+^{|A|}$ contenant les valeurs de toutes les capacités.
On associe à chaque arête (u, v) un flot f_{uv} représentant la quantité de flot vérifiant deux contraintes :

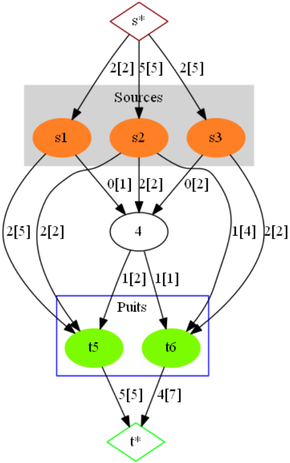
1. Contrainte de capacité : $f_{uv} \leq c_{uv}$ pour toute arête $(u, v) \in A$;
2. Conservation du flot : $\sum_{u:(u,v)\in A} f_{uv} = \sum_{u:(v,u)\in A} f_{vu}$ pour tout

$v \in V \setminus s, t$. Cette contrainte s'appelle aussi la loi de Kirchhoff.
La valeur du flot représente la quantité de flot allant de la source au puit. Elle se définie par $|f| = \sum_{v \in V} f_{sv}$.

Problème de flot max

1. Les Sources sont reliées à une Source Mère, avec un arc mère-fille de capacité équivalent à la production de la source.
2. Les Puits sont reliés à un Puit Père, avec un arc fils-père de capacité équivalent à la consommation du puit.
3. Ford-Fulkerson fournit une configuration de flot max compatible avec l'offre et la demande.

Graphe classique



Ford-Fulkerson

Initialisation par un flot initial réalisable ($f = 0$)
Tant que le flot n'est pas maximal
- Marquage de la source $s*$
- Tant qu'on marque des sommets
  - Pour tout sommet marqué i
    - Marquer les sommets j non marqués tel que $f(i,j) < c(i,j)$ ou $f(j,i) > 0$
  - Fin Pour
- Fin Tant que
- Si le puit $t*$ n'est pas marqué alors le flot est maximal
- Sinon Amélioration du flot()

Fin Tant que

Amélioration du flot

Trouver une chaîne qui a permis de marquer $t*$ et calculer

$\varepsilon = min(\varepsilon_1, \varepsilon_2) > 0$ avec

$\varepsilon_1 = min\{c(u,v) - f(u,v)$ avec $(u,v) \in A$ (arête directe)$\}$

$\varepsilon_1 = min\{f(v,u)$ avec $(u,v) \in A$ (arête inverse)$\}$

Trouver le nouveau flot f' dans le chemin choisi

Si $(u,v) \in A$ alors $f'(u,v) = f(u,v) + \varepsilon$

Sinon $f'(u,v) = f(u,v) - \varepsilon$ (arête inverse)

Lignes directrices

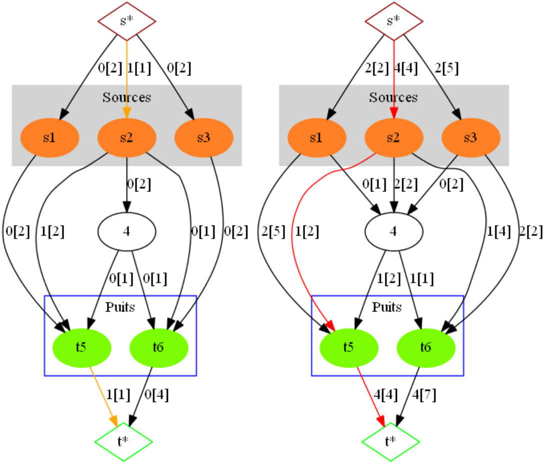
1. Niveau local
   - Attentes et besoins
   - Résolution par KP
   - Exemple
2. Niveau microgrid
   - Attentes et besoins
   - Pseudo-enchères
   - Equilibre du système
3. Niveau T&D
   - Attentes et besoins
   - Module d'ajustement
   - Equilibre de l'offre et la demande

Module d'ajustement

Pour tout $(u, v) \in A$
    Si $f(u, v)_{(t_{i-1})} > c(u, v)_{(t_i)}$
        $c* = f(u, v)_{(t_{i-1})} - c(u, v)$
        Ajouter $((u, v), c*)$ à la Liste
    Fin Si
Fin Pour Tout

1. En cas de différence « à la baisse » de la production ou de la consommation, on effectue le module d'ajustement.
2. Ford-Fulkerson sur le module d'ajustement, le flot max est soustrait du résultat de l'itération précédente.
3. Ford-Fulkerson sur le graphe « ajusté ».

Ajustement


s*
0[2]
1[1]
0[2]
Sources
s1
s2
s3
0[2]
0[2]
1[2]
4
0[1]
0[2]
0[1]
0[1]
Puits
t5
t6
1[1]
0[4]
t*
s*
2[2]
4[4]
2[5]
Sources
s1
s2
s3
0[1]
2[2]
0[2]
2[5]
1[2]
4
1[4]
2[2]
1[2]
1[1]
Puits
t5
t6
4[4]
4[7]
t*

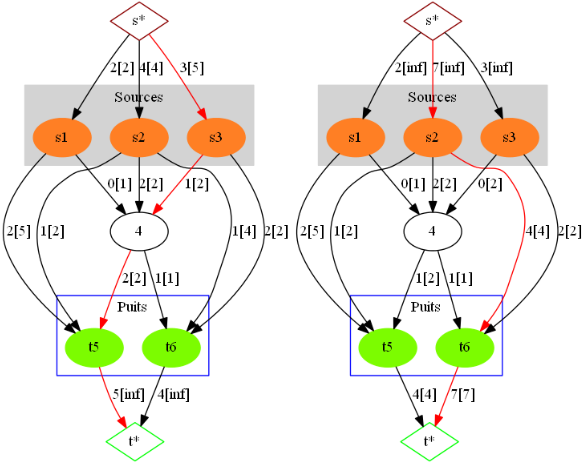
Lignes directrices

Graphe évolutif

Pour cela nous effectuons deux tests :

1. Ford-Fulkerson sur le graphe classique, sans contrainte de capacité sur les arcs $(t, t*)$. Le résultat fournira le flot maximum sans contrainte de consommation.
2. Ford-Fulkerson sur le graphe classique, sans contrainte de capacité sur les arcs $(s*, s)$. Le résultat permettra de tenir compte de la demande de consommation, et ainsi de mieux prévoir la consommation future.

Résultats


s*
2[2]
4[4]
3[5]
Sources
s1
s2
s3
0[1]
2[2]
1[2]
2[5]
1[2]
4
1[4]
2[2]
2[2]
1[1]
Puits
t5
t6
5[inf]
4[inf]
t*
s*
2[inf]
7[inf]
3[inf]
Sources
s1
s2
s3
0[1]
2[2]
0[2]
2[5]
1[2]
4
4[4]
2[2]
1[2]
1[1]
Puits
t5
t6
4[4]
7[7]
t*

Rétroaction

Les résultats des graphes à sources ou puits de capacités infinies permettent une rétroaction.

1. Définition des puits sous ou sur consommateur :
   1. Mise à jour des pronostiques ;
   2. Mise à jour des enchères.
2. Définition des sources sous ou sur producteur :
   1. Mise à jour des pronostiques ;
   2. Planification de la production au temps prochain.

Avantages et Inconvénients

1. Avantages :
   1. Résolution rapide et satisfaisant l'offre et la demande ;
   2. Calcul rapide pour l'évolution du système.
2. Inconvénients :
   1. Dépendant de l'initialisation du système et des premières instances ;
   2. Graphe classique, non prétopologique.

Résumé

- Niveau local : KP et gestion de la domotique.
- Niveau microgrid : pseudo-enchères et KP microgrid.
- Niveau T&D : Ford-Fulkerson, graphe d'évolution et gestion des prognostiques.

- Perspectives :
  - Modifier les speudo-enchères en jeux de coopération ou d'enchères généralisées ;
  - Généralisation du réseau de jeux ;
  - Modifier Ford-Fulkerson pour le rendre compatible à la prétopologie ;
  - Création des algorithmes pour tenir compte de l'économie de marché T&D ;
  - Gestion des données et statistiques.