

MODÈLE SIMPLE D'OPTIMISATION DE RÉSEAUX D'ANTENNES

Résumé : Dans ce texte, nous introduisons un modèle simple d'optimisation de réseaux d'antennes. Ce modèle fait apparaître naturellement des matrices ayant une structure particulière pour lesquelles différents algorithmes plus efficaces que les méthodes usuelles peuvent être utilisés.

Mots clefs : Algèbre linéaire. Méthodes itératives. Transformée de Fourier discrète.

- *Il est rappelé que le jury n'exige pas une compréhension exhaustive du texte. Vous êtes laissé(e) libre d'organiser votre discussion comme vous l'entendez. Des suggestions de développement, largement indépendantes les unes des autres, vous sont proposées en fin de texte. Vous n'êtes pas tenu(e) de les suivre. Il vous est conseillé de mettre en lumière vos connaissances à partir du fil conducteur constitué par le texte. Le jury appréciera que la discussion soit accompagnée d'exemples traités sur ordinateur.*

1. Le modèle d'antennes

Nous nous intéressons ici à un modèle simple de réseau et d'antennes. Une antenne est un dispositif émetteur d'ondes électromagnétiques. On appelle **puissance nominale** de l'antenne la valeur x de la puissance émise par l'antenne au point où elle se trouve. La puissance reçue par un récepteur situé à une distance r de l'antenne est alors donnée par $xV(r)$ où $V(r)$ est une fonction normalisée (i.e. $V(0) = 1$) qui mesure la perte de puissance du signal en fonction de la distance. On supposera dans la suite qu'en première approximation cette perte de signal est bien décrite en prenant $V(r) = \frac{1}{1 + \left(\frac{r}{r_0}\right)^2}$, où $r_0 > 0$ est une distance caractéristique du modèle.

Dans le modèle unidimensionnel de réseau que nous proposons d'étudier, nous supposons que n antennes comme ci-dessus sont situées sur une droite et que deux antennes successives sont distantes d'une longueur d donnée et fixée. Les positions des antennes sont repérées par les abscisses $a_i = id$ pour $i \in \{1, \dots, n\}$.

Si on note x_i la puissance nominale de l'antenne numéro i , la **puissance effective** reçue par un récepteur situé à la position $s \in \mathbb{R}$ est donnée par

$$(1) \quad \Pi(s) = \sum_{i=1}^n x_i V(|s - a_i|).$$

On s'intéresse plus particulièrement à la puissance effective du signal aux pieds des antennes. Ainsi, pour un vecteur $x = (x_i)_{1 \leq i \leq n}$ des puissances nominales donné, le vecteur $\pi = (\pi_i)_{1 \leq i \leq n}$ des puissances effectives au pied de chaque antenne est donné par

$$(2) \quad \pi_i = \Pi(a_i) = \sum_{j=1}^n \frac{x_j}{1 + \frac{d^2}{r_0^2} |i - j|^2}.$$

2. Le problème du réseau complet

2.1. Position du problème

On souhaite maintenant déterminer les valeurs que l'on doit donner aux puissances nominales $x = (x_i)_{1 \leq i \leq n}$ des différentes antennes pour que les puissances effectives $(\pi_i)_{1 \leq i \leq n}$ au pied des antennes soient toutes égales à la même valeur, que l'on prendra égale à 1 pour fixer les idées.

On note donc $\pi = {}^t(1, \dots, 1)$, et on cherche un vecteur de réels positifs $x \in \mathbb{R}^n$ tel que

$$(3) \quad \pi = Ax, \quad \text{avec} \quad A = \begin{pmatrix} 1 & V(d) & V(2d) & \dots & V((n-1)d) \\ V(d) & 1 & V(d) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & V(2d) \\ \vdots & \ddots & \ddots & 1 & V(d) \\ V((n-1)d) & \dots & \dots & V(d) & 1 \end{pmatrix}.$$

Pour tout $\gamma > 0$, on pose $f(\gamma) = \sum_{m=1}^{\infty} \frac{1}{1 + \gamma^2 m^2}$. On note γ_0 et γ_1 les uniques réels positifs tels que $f(\gamma_0) = \frac{1}{2}$ et $f(\gamma_1) = \frac{1}{4}$.

Théorème 1. Si $\frac{d}{r_0} > \gamma_0$, alors la matrice $A \in M_n(\mathbb{R})$ est définie positive pour toute valeur de n et dans ce cas, l'unique solution x de (3) vérifie

$$(4) \quad \sup_{1 \leq i \leq n} |x_i| \leq \frac{1}{1 - 2f\left(\frac{d}{r_0}\right)}.$$

Théorème 2. Si $\frac{d}{r_0} \geq \gamma_1$, alors l'unique solution x de (3) est positive. Plus précisément, on a :

$$(5) \quad \inf_{1 \leq i \leq n} x_i \geq \frac{1 - 4f\left(\frac{d}{r_0}\right)}{1 - 2f\left(\frac{d}{r_0}\right)}.$$

Sous de bonnes hypothèses, le problème (3) admet donc une unique solution positive. Si on utilise la méthode LU (ou pivot de Gauss) pour résoudre ce problème, cela va coûter $O(n^3)$ opérations, ce qui n'est pas très raisonnable pour n grand. On se propose de trouver des algorithmes efficaces qui utiliseront la structure particulière de la matrice.

Définition 1. Une matrice T carrée d'ordre n à coefficients réels sera dite de Toeplitz symétrique si et seulement s'il existe un vecteur de \mathbb{R}^n noté $t = (t_k)_{k \in \{0, \dots, n-1\}}$ tel que

$$\text{pour tout } (i, j) \text{ dans } \{1, \dots, n\}^2 \text{ tel } |j - i| = k, \quad \text{on a } T_{i,j} = t_k.$$

Notons que le vecteur t n'est autre que la première colonne de la matrice T .

2.2. Solveur direct

On présente ici une méthode directe en deux étapes pour résoudre un système linéaire dont la matrice T est Toeplitz symétrique définie positive. Pour tout $k \in \{1, \dots, n\}$, on note T_k la matrice carrée de taille k obtenue en conservant seulement les k premières lignes et k premières colonnes de T . On note S_k l'opérateur d'inversion des indices pour les vecteurs de \mathbb{R}^k défini par

$$(6) \quad \forall y \in \mathbb{R}^k, \quad \forall i \in \{1, \dots, k\}, \quad (S_k y)_i = y_{k+1-i}.$$

- **Etape 1 :** On construit une famille de vecteurs f_k de taille k pour tout $k = 1, \dots, n$ de la façon suivante :

— On pose $f_1 = \begin{pmatrix} 1 \\ t_0 \end{pmatrix} \in \mathbb{R}$.

— Pour $k = 2, \dots, n$, on détermine $\alpha_k, \beta_k \in \mathbb{R}$ tels que

$$\text{le vecteur } f_k = \alpha_k \begin{pmatrix} S_{k-1} f_{k-1} \\ 0 \end{pmatrix} + \beta_k \begin{pmatrix} 0 \\ f_{k-1} \end{pmatrix} \text{ vérifie } T_k f_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

- **Etape 2 :** Soit à résoudre $Tx = b$. On construit une famille de vecteurs x_k de taille k pour tout $k = 1, \dots, n$ vérifiant $T_k x_k = {}^t(b_1, \dots, b_k)$. Pour cela on effectue la récurrence suivante :

— On pose $x_1 = \begin{pmatrix} b_1 \\ t_0 \end{pmatrix}$.

— Pour tout $k = 2, \dots, n$, on cherche x_k sous la forme

$$(7) \quad x_k = \begin{pmatrix} x_{k-1} \\ 0 \end{pmatrix} + \theta_k f_k,$$

où $\theta_k \in \mathbb{R}$ est déterminé pour que l'équation $T_k x_k = {}^t(b_1, \dots, b_k)$ soit vérifiée.

Cet algorithme nécessite $O(n^2)$ opérations, il est donc plus rapide que la méthode du pivot de Gauss pour des grandes valeurs de n .

3. Le problème du réseau partiel

3.1. Position du problème

On s'intéresse à la situation dans laquelle un certain nombre d'antennes du réseau sont en panne. On notera I l'ensemble des indices $i \in \{1, \dots, n\}$ correspondant aux antennes qui fonctionnent et P la matrice de projection sur cet ensemble I (i.e. $p_{ij} = 1$ si $i = j \in I$ et $p_{ij} = 0$

sinon). On souhaiterait donc résoudre l'équation (3) sous la contrainte $Px = x$, ce qui n'est bien sûr pas possible en général.

Il nous faut donc relaxer le problème. Nous proposons deux façons de le faire.

- **Première approche :** On demande que la puissance effective soit exactement égale à 1 uniquement au pied des antennes qui fonctionnent. On se propose donc de résoudre

$$(8) \quad \begin{cases} PAPx = P\pi, \\ \text{avec } Px = x. \end{cases}$$

- **Seconde approche :** On essaie de faire en sorte que les puissances effectives au pied de toutes les antennes soient en moyenne les plus proches de 1. Par exemple, on peut chercher un vecteur x qui réalise le problème d'optimisation suivant

$$(9) \quad \inf_{x \in \mathbb{R}^n, Px=x} \|Ax - \pi\|_2^2,$$

où $\|\cdot\|_2$ est la norme euclidienne usuelle de \mathbb{R}^n . On vérifie que ceci revient à chercher l'unique solution du problème linéaire suivant

$$(10) \quad \begin{cases} P^t AAPx = P^t A\pi, \\ \text{avec } Px = x. \end{cases}$$

3.2. Solveur itératif

Dans le cadre du réseau partiel, les matrices qui interviennent sont PAP ou $P^t AAP$ qui, malheureusement, n'ont plus la structure Toeplitz symétrique de la matrice A . On ne peut donc pas utiliser de façon immédiate le solveur direct proposé dans le paragraphe 2.2.

Pour résoudre un problème du type $PMPx = Pb$ avec $Px = x$ où M est une matrice symétrique définie positive et P la projection définie plus haut, on propose d'utiliser une version adaptée de l'algorithme du gradient conjugué qui s'écrit

— **Initialisation :** Choisir x_0 tel que $Px_0 = x_0$. Calculer $\rho_0 = Pb - PMPx_0$. Poser $g_0 = \rho_0$.

— **Pour $k = 0, \dots$, faire :**

$$\alpha_k = \frac{\langle \rho_k, \rho_k \rangle}{\langle PMPg_k, g_k \rangle}, \quad x_{k+1} = x_k + \alpha_k g_k, \quad \rho_{k+1} = \rho_k - \alpha_k PMPg_k,$$

$$\beta_k = \frac{\langle \rho_{k+1}, \rho_{k+1} \rangle}{\langle \rho_k, \rho_k \rangle}, \quad g_{k+1} = \rho_{k+1} + \beta_k g_k.$$

On possède alors le théorème de convergence suivant :

Théorème 3. *L'algorithme proposé converge en au plus n itérations vers l'unique solution de $PMPx = Pb$, $Px = x$.*

Chaque itération de l'algorithme nécessite un certain nombre de calculs de produits scalaires mais surtout un calcul d'un produit matrice/vecteur de la forme My . Dans le cadre qui nous intéresse, la matrice M (égale à A ou ${}^t AA$) est pleine et le calcul d'un tel produit matrice/vecteur coûte donc *a priori* $O(n^2)$ opérations.

Au final, le solveur itératif proposé va coûter, au pire, $O(n^3)$ opérations.

3.3. Produit matrice/vecteur pour une matrice Toeplitz

Dans le but d'obtenir une méthode moins coûteuse, nous allons voir comment utiliser la structure Toeplitz symétrique des matrices qui nous occupent ici pour effectuer le calcul d'un produit matrice/vecteur en $O(n \log n)$ opérations.

Pour ce faire, on commence par remarquer que toute matrice Toeplitz symétrique T peut être plongée dans une matrice circulante symétrique de taille $2n$ (notée $C(T)$) comme dans l'exemple suivant : Soit $t = {}^t(t_0, t_1, t_2)$ et T la matrice de Toeplitz engendrée par t , alors

$$(11) \quad C(T) = \begin{pmatrix} t_0 & t_1 & t_2 & 0 & t_2 & t_1 \\ t_1 & t_0 & t_1 & t_2 & 0 & t_2 \\ t_2 & t_1 & t_0 & t_1 & t_2 & 0 \\ 0 & t_2 & t_1 & t_0 & t_1 & t_2 \\ t_2 & 0 & t_2 & t_1 & t_0 & t_1 \\ t_1 & t_2 & 0 & t_2 & t_1 & t_0 \end{pmatrix} = \begin{pmatrix} T & D \\ D & T \end{pmatrix}, \quad D \in \mathcal{M}_n(\mathbb{R}).$$

Un produit matrice/vecteur Ty peut s'obtenir en calculant le produit $C(T) \begin{pmatrix} y \\ 0 \end{pmatrix}$ et en ne conservant que les n premières coordonnées. Introduisons la matrice \mathcal{F}^m de la transformée de Fourier discrète, définie par

$$(12) \quad \forall i, j \in \{1, \dots, m\}, \quad \mathcal{F}_{ij}^m = e^{\frac{2\pi}{m}\sqrt{-1}(i-1)(j-1)}.$$

Théorème 4. Soit C une matrice symétrique circulante de taille m dont on note $c \in \mathbb{R}^m$ le premier vecteur colonne. On a

$$(13) \quad C = \mathcal{F}^m \text{diag}(\mathcal{F}^m c) (\mathcal{F}^m)^{-1},$$

où, pour $v \in \mathbb{R}^m$, $\text{diag}(v)$ désigne la matrice diagonale dont la diagonale est donnée par v .

Ainsi, pour calculer le produit matrice/vecteur de C par un vecteur de \mathbb{R}^m , on est ramené à calculer des produits matrice/vecteur pour la matrice \mathcal{F}^m et son inverse ainsi que le produit par une matrice diagonale. Or l'algorithme de **Transformation de Fourier Rapide** (ou FFT) permet de calculer de tels produits en $O(m \log m)$ opérations.

n	200	500	1000	2000
calcul direct	0.13	2	20.49	164.6
calcul rapide	0.03	0.13	0.5	2.21

TABLE 1. Temps de calcul (en secondes) de n produits matrice/vecteur

En conclusion, on peut mettre en place une version de l'algorithme du gradient conjugué pour résoudre les problèmes (8) ou (10) qui coûte, au pire, $O(n^2 \log n)$ opérations. De plus, la mise en place de cet algorithme ne nécessite pas la construction explicite (et donc le stockage en mémoire) de la matrice T . Le tableau 1 donne une idée des temps de calculs obtenus (sur un ordinateur personnel) en effectuant, pour différentes valeurs de n , le calcul de n produits matrice/vecteur pour une matrice Toeplitz de taille n .

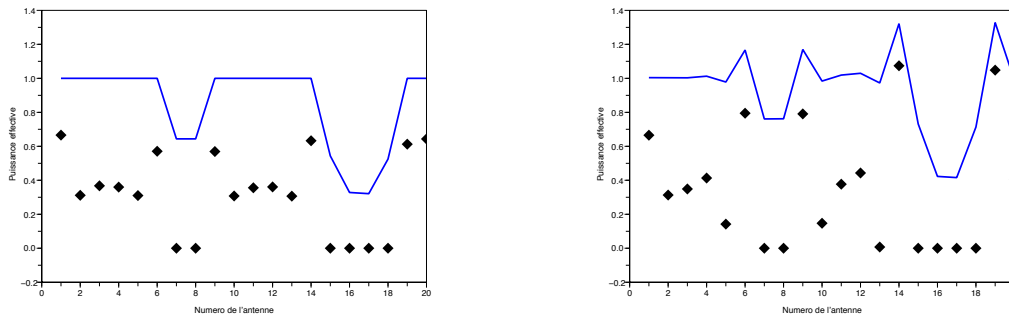


FIGURE 1. Résultats pour la première approche (à gauche) et la seconde (à droite)

4. Exemple numérique

On considère ici le cas d'un réseau de 20 antennes avec $d = r_0$. On se place dans le cas du réseau partiel où les antennes numéros 7, 8, 15, 16, 17, 18 sont en panne. Les résultats obtenus par les deux approches (8) et (10) sont montrés dans la figure 1. Les points symbolisés par \blacklozenge représentent les puissances nominales des antennes calculées en résolvant le système linéaire. La courbe en trait continu donne la répartition de la puissance effective au pied de chacune des antennes pour ces valeurs des puissances nominales.

Suggestions pour le développement

- *Soulignons qu'il s'agit d'un menu à la carte et que vous pouvez choisir d'étudier certains points, pas tous, pas nécessairement dans l'ordre, et de façon plus ou moins fouillée. Vous pouvez aussi vous poser d'autres questions que celles indiquées plus bas. Il est très vivement souhaité que vos investigations comportent une partie traitée sur ordinateur et, si possible, des représentations graphiques de vos résultats.*
 - Démontrer tout ou partie des théorèmes énoncés dans le texte.
 - Discuter l'algorithme de la section 2.2 (calcul de α_k , β_k , θ_k , coût de calcul, stockage, ...) et montrer qu'il fournit bien le résultat annoncé.
 - Montrer que les problèmes (8) et (10) ont une unique solution. Montrer que le problème (10) est bien équivalent au problème d'optimisation (9).
 - Comment pourrait-on affecter des poids plus grands à certaines antennes (jugées plus importantes que les autres) dans le problème de minimisation (9) ?
 - Discuter la validité de l'algorithme de type *gradient conjugué* proposé dans le texte.
 - Détailler les calculs de la section 3.3 et tester numériquement la méthode proposée. On pourra utiliser les routines de transformée de Fourier rapide disponibles dans les logiciels de calcul à votre disposition.
 - Vérifier les résultats de la figure 1 dans le cadre de l'exemple proposé. On pourra utiliser les algorithmes du texte ou, à défaut, les fonctionnalités des logiciels à votre disposition.