Modèles et algorithmes des réseaux

DM 1 : date limite 4 décembre 2018

Exercice 1. Routage algébrique

Dans cet exercice on étudie une généralisation du routage algébrique au cas de différents domaines avec des métriques parfois différentes. Comme en cours, le réseaux à l'intérieur d'un domaine est représenté par un graphe dirigé $G = (V, E)$, et les poids des routes sont modélisés par un semi-anneau $(S, \oplus, \otimes)$ et une fonction $w : E \to S$. Nous allons noter par 0_S l'élément nul de $(S, \oplus)$ et par 1_S l'élément neutre de $(S, \otimes)$. On note A la matrice d'incidence induite par w. Dans la suite, on suppose que A^* existe, et on notera $|V| = n$.

On considère maintenant un ensemble de nœuds externes D (appelé l'ensemble des destinations), disjoint de V. Les nœuds dans D sont attachés au graphe G via des arcs externes $E' \subset V \times D$ (correspondant à des chemins qui passent par un autre domaine). Nous allons d'abord considérer le cas ou les poids des arcs dans E' sont également dans $(S, \oplus, \otimes)$. Soit M la matrice d'incidence, de taille $|V| \times |D|$, pour les arcs externes.

Q1. On pose $F = A^* \otimes M$. Montrer que F est une solution de l'équation

$$F = (A \otimes F) \oplus M.$$

À quoi correspond la matrice F pour $(S, \oplus, \otimes) = (\mathbb{R}_{\min}, \min, +)$?

Par la suite, nous allons supposer que les poids des arcs externes sont donnés par une fonction w' : $E' \to N$, où $(N, \Box)$ est un monoïde commutatif, et on note par 0_N son élément neutre. Donc la matrice M est dans $M_{n,m}(N)$, avec $m = |D|$. Afin de pouvoir combiner les poids dans S avec des poids dans N, nous allons utiliser une nouvelle opération $\rhd$: $S \times N \to N$. Nous allons supposer que $(N, \Box, \rhd)$ est un semi-module, i.e. que les opérations $\Box$ et $\rhd$ vérifient :

(1) $x \rhd (a \Box b) = (x \rhd a) \Box (x \rhd b),\ \forall x \in S,\ \forall a, b \in N$;
(2) $(x \oplus y) \rhd a = (x \rhd a) \Box (y \rhd a),\ \forall x, y \in S,\ \forall a \in N$;
(3) $(x \otimes y) \rhd a = x \rhd (y \rhd a)$;
(4) $0_S \rhd a = 0_N,\ \forall a \in N$;
(5) $x \rhd 0_N = 0_N,\ \forall x \in S$;
(6) $1_S \rhd a = a,\ \forall a \in N$.

Pour les matrices $M, M' \in M_{n,d}$, et $R \in M_n(S)$, nous pouvons alors définir les opérations $\Box$ et $\rhd$ comme suit :

$$(M \Box M')_{i,j} = M_{i,j} \Box M'_{i,j},$$

et

$$(R \rhd M)_{i,d} = \Box_{1 \le k \le n}(R_{i,k} \rhd M_{k,d}).$$

Q2. Soit $(N, \Box, \rhd)$ un semi-module sur un semi-anneau $(S, \oplus, \otimes)$. On supposé que l'operation $\Box$ est idémpotente. On considère l'équation suivante, d'inconnue $X \in M_n(S)$:

$$X = (A \otimes X) \oplus I,$$

où I est la matrice identité dans $M_n(S)$. Montrer que pour toute solution R de cette équation, $R \rhd M$ est une solution de l'équation :

$$F = (A \rhd F) \Box M.$$

Soit $(S, \oplus_S, \otimes_S)$ un semi-anneau intègre et sélectif (i.e. l'opération $\oplus_S$ est sélective) et $(T, \oplus_T)$ un monoïde. On définit le produit lexicographique gauche le monoïde $S \overrightarrow{\times} T = ((S \times T) \cup \{\infty\}, \overrightarrow{\oplus})$, avec l'élément neutre ∞, où pour $(s,t), (s',t') \in S \backslash \{0_S\} \times T \backslash \{0_T\}$,

$$(s,t) \overrightarrow{\oplus} (s',t') = \begin{cases} (s, t \oplus_T t') & s = s \oplus_S s' = s' \\ (s,t) & s = s \oplus_S s' \neq s' \\ (s',t') & s \neq s \oplus_S s' = s', \end{cases}$$

et $(0_S, t) = (s, 0_T) = \infty,\ \forall s \in S, t \in T$.

On définit $\rhd_f$: $S \to (S \times T) \cup \{\infty\} \times (S \times T) \cup \{\infty\}$ comme suit :

$$s \rhd_f (s', t) = (s \otimes_S s', t), \quad \text{et} \quad s \rhd_f \infty = \infty.$$

Q3. Montrer que $\mathrm{Hot}(S,T) = ((S \times T) \cup \{\infty\}, \overrightarrow{\oplus}, \rhd_f)$ est un semi-module.

Le semi-module $\mathrm{Hot}(S,T)$ donne la priorité aux routes les plus courtes (au sens de semi-anneau $(S, \oplus_S, \otimes_S)$), puis les égalités sont partagés par des poids dans $(T, \oplus_T)$. Ce routage est également appelé “hot potato routing”.

Q4. On suppose maintenant que $(T, \oplus_T)$ est un monoïde sélectif. Donner un modèle algébrique du routage $\mathrm{Cold}(S,T)$ qui donne la priorité aux poids dans $(T, \oplus_T)$, avant de considerer les poids dans $(S, \oplus_S, \otimes_S)$? S'agit-il d'un semi-module ?

Q5. Soient $(S, \oplus_S, \otimes_S) = (\mathbb{R}_{\min}, \min, +)$, $(T, \oplus_T) = (\mathbb{R}_{\min}, \min)$,

$$A = \begin{pmatrix} \infty & 2 & 1 & 6 & \infty \\ 2 & \infty & 5 & \infty & 4 \\ 1 & 5 & \infty & 4 & 3 \\ 6 & \infty & 4 & \infty & \infty \\ \infty & 4 & 3 & \infty & \infty \end{pmatrix}, \quad \text{et} \quad M = \begin{pmatrix} \infty & \infty \\ 3 & \infty \\ \infty & \infty \\ \infty & 1 \\ 2 & 3 \end{pmatrix}.$$

Calculer les matrices $R \rhd M$ pour les semi-modules $\mathrm{Hot}(S,T)$ et $\mathrm{Cold}(S,T)$. Comparer avec F de la question Q1 pour $(S, \oplus, \otimes) = (\mathbb{R}_{\min}, \min, +)$.

Supposons maintenant que les destinations peuvent être connectées par trois types différents des arcs externes, types 0, 1, et 2. Les arcs externes du type 0 sont préférés devant types 1 ou 2 et les arcs externes du type 1 sont préférés devant ceux du type 2. On note par $(S, \oplus_S, \otimes_S)$ est le semi-anneau sélectif utilisé pour les poids des chemins internes dans G, et par $(T_i, \oplus_{T_i})$ le monoïde utilisé pour les poids des arcs externes du type $i \in \{0,1,2\}$. Pour les arcs externes du type 0 et 1, la priorité est donnée aux poids des chemins du graphe G, et pour les arcs du type 2, aux poids externes. Donc, on suppose que $(T_2, \oplus_{T_2})$ est un monoïde.

Q6. Donner un semi-module $(N, \square, \rhd)$ pour ce routage. Illustrer la matrice $R \rhd M$ pour $(S, \oplus_S, \otimes_S) = (\mathbb{R}_{\min}, \min, +)$, $(T_i, \oplus_{T_i}) = (\mathbb{R}_{\min}, \min)$ pour $i \in \{0,1\}$, et $(T_2, \oplus_{T_2}) = (\mathbb{R}_{\min}, \max)$, avec

$$A = \begin{pmatrix} \infty & 2 & 1 & 6 & \infty \\ 2 & \infty & 5 & \infty & 4 \\ 1 & 5 & \infty & 4 & 3 \\ 6 & \infty & 4 & \infty & \infty \\ \infty & 4 & 3 & \infty & \infty \end{pmatrix}, \ M_0 = \begin{pmatrix} \infty & \infty \\ \infty & \infty \\ \infty & \infty \\ \infty & \infty \\ 2 & \infty \end{pmatrix}, \ M_1 = \begin{pmatrix} \infty & \infty \\ 3 & \infty \\ \infty & \infty \\ \infty & \infty \\ 17 & \infty \end{pmatrix}, \ \text{et } M_2 = \begin{pmatrix} \infty & \infty \\ 4 & \infty \\ \infty & 1 \\ \infty & \infty \\ 1 & 2 \end{pmatrix}.$$

Exercice 2. Protocole Push-Sum

Le protocole Gossip, dont une implémentation a déjà été étudiée en cours, est un protocole de calcul distribué dans un réseau. On considère ici une de ses versions, le protocole Push-Sum, qui se déroule de la façon suivante.

Chaque sommet i du graphe maintient à chaque temps t deux valeurs : une *somme* $s_{t,i}$ et un *poids* $w_{t,i}$. Au temps 0, on suppose que $s_{0,i} = x_i$ et $w_{0,i}$; par la suite, à chaque étape t, chaque sommet i suit l'algorithme suivant : on note $\{(\hat{s}_r, \hat{w}_r)\}$ les paires reçues par i au temps $t-1$ (on suppose qu'au temps 0 le sommet i a reçu la paire $(s_{0,i}, w_{0,i})$).

— calculer $s_{t,i} = \sum_r \hat{s}_r$ et $w_{t,i} = \sum_r \hat{w}_r$.

— choisir au hasard un de ses voisins $f_t(i)$.

— envoyer la paire $(\frac{1}{2}s_{t,i}, \frac{1}{2}w_{t,i})$ à $f_t(i)$ et à lui-même.

L'*estimation* d'un sommet i au temps t est la valeur $\frac{s_{t,i}}{w_{t,i}}$.

Q1. En supposant que l'algorithme converge vers une même valeur pour chaque sommet, quelle est cette valeur ?

Q2. Implémenter (en Python) l'algorithme Push-Sum décrit ci-dessus. Le tester sur des graphes de densité variable, et tracer l'erreur relative

$$e_t = \frac{1}{\sum_j |x_j|} \max_i \left| \frac{s_{t,i}}{w_{t,i}} - \ell \right|$$

en fonction du temps et de la densité, où ℓ est la limite trouvée à la question précédente. Que remarque-t-on ?

Q3. Comment modifier l'algorithme ci-dessus pour calculer le nombre de sommets d'un réseau de façon distribuée ? Implémenter cette modification, et en étudier la convergence.

Exercice 3. Protocole ALOHA

On considère dans cet exercice le protocole de communication ALOHA, mis en place à l'université de Hawaii dans les années 1970. Dans la version discrète du protocole, plusieurs stations essaient ou non d'envoyer des messages via un même canal à intervalles réguliers (ou *slots*). S'il y a collision (i.e. si plusieurs messages essaient d'être transmis pendant le même *slot*), la transmission est annulée et les messages devront être retransmis ultérieurement.

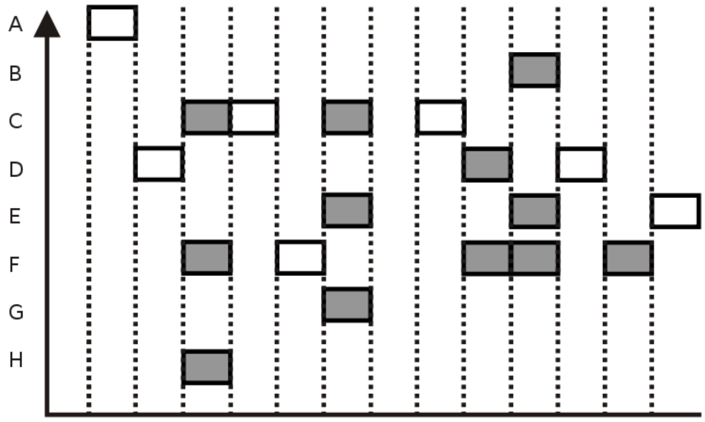

Figure 1. Exemple de fonctionnement du protocole ALOHA. Chaque rectangle correspond à un message envoyé sur le canal, et les messages grisés n'ont pas été transmis pour cause de collision.

Q1. On suppose tout d'abord que le nombre de messages envoyés à chaque instant suit une loi $\mathrm{Poi}(\lambda)$. Quel est le débit moyen (i.e. le nombre moyen de messages transmis par unité de temps) du système ?

On souhaite maintenant modéliser le protocole ALOHA via une chaîne de Markov. À cet effet, on note A_n le nombre de messages envoyés sur le canal à l'étape n, et on suppose que les variables A_n sont i.i.d de loi

$$\mathbb{P}(A_n = j) = a_j,$$

et d'espérance λ. On suppose de plus que les messages non transmis sont stockés dans une file à part, et que chaque message de la file est transmis indépendamment avec probabilité ν à chaque étape. On note X_n la taille de la file d'attente à la fin de l'étape n. On définit aussi les quantités

$$\alpha_n = \mathbb{P}(\mathrm{Bin}(n,\nu) = 0) = (1-\nu)^n \quad \text{et} \quad \beta_n = \mathbb{P}(\mathrm{Bin}(n,\nu) = 1) = n\nu(1-\nu)^{n-1}$$

Q2. Montrer que la suite $(X_n)_{n\in\mathbb{N}}$ est une chaîne de Markov, et donner son noyau de transition. Sous quelle condition cette matrice est-elle irréductible ? Apériodique ?

Q3. [Code] Simuler la chaîne X dans le cas où $A_n \sim \mathrm{Poi}(\lambda)$. La fonction prendra en argument le paramètre λ ainsi que la probabilité de retransmission ν. Qu'observe-t-on ?

Q4. On suppose par l'absurde que la chaîne X est récurrente, et qu'elle a donc une distribution stationnaire π. Montrer que π satisfait l'équation

$$\begin{aligned}\pi(i) = \pi(i)\left((1-\beta_i)\,a_0 + \alpha_i a_1\right) + \beta_{i+1} a_0 \pi(i+1) \\ + (1-\alpha_{i-1}) a_1 \pi(i-1) + \sum_{\ell=2}^{\infty} \pi(i-\ell) a_\ell,\end{aligned}$$

où $\pi(i) = 0$ si $i < 0$.

Q5. En posant
$$P_N = \sum_{i \leq N} \pi(i),$$
montrer la relation suivante :
$$P_N = \pi(N)\alpha_N a_1 + \pi(N+1)\beta_{N+1}a_0 + \sum_{\ell=0}^{N} a_\ell P_{N-\ell}$$

Q6. En déduire que
$$\frac{\pi(N+1)}{\pi(N)} \geq \frac{1 - a_0 - \alpha_N a_1}{\beta_{N+1}a_0}$$
et en tirer une contradiction.

On cherche maintenant à stabiliser le protocole ; une idée naturelle est de faire dépendre le paramètre ν du nombre de messages en attente ; on notera donc $\nu = \nu(k)$. On cherche à déterminer la région de stabilité maximale du processus.

Q7. En utilisant le lemme de Pake (TD du 16/10), en déduire qu'une condition suffisante de stabilité est
$$\lambda < \liminf_{i \to +\infty} g_i(\nu(i)),$$
où $g_k(\nu) = (1-\nu)^k a_1 + k\nu(1-\nu)^{k-1}a_0$.

Q8. On suppose que $a_0 > a_1$. Montrer que le choix optimal de ν est
$$\nu(k) = \frac{a_0 - a_1}{ka_0 - a_1},$$
et en déduire une région de stabilité du protocole ALOHA.

Q9. [Code] Simuler des trajectoires de X_n dans le cas stable. Qu'observe-t-on par rapport à la simulation précédente ?

Exercice 4. Algorithme MaxWeight généralisé

Dans cet exercice on étudie une généralisation de MaxWeight. On considère un système constitué de n serveurs (stations, utilisateurs, ressources, etc...). Le temps est discrétisé, et à chaque unité de temps un certain nombre de paquets arrive aux serveurs : pour $t \in \mathbb{N}$, le vecteur $\mathbf{a}_t = (a_t(k), k = 1, \ldots, n) \in \mathbb{N}^n$ enregistre le nombre de paquets qui arrivent aux différents serveurs à l'instant t. On suppose les arrivées i.i.d. au cours du temps, i.e., si $\mathbf{a}_t = (a_t(k), k = 1, \ldots, n)$ avec $a_t(k)$ le nombre d'arrivées au serveur k à l'instant t, alors les $(\mathbf{a}_t, t \geq 0)$ sont i.i.d., et on note $\lambda_k = E(a_1(k))$ et $\lambda = (\lambda_1, \ldots, \lambda_n)$. Chaque serveur est doté d'une file d'attente, dans laquelle les paquets sont stockés en attendant d'être servis.

Un vecteur $\sigma \in \{0,1\}^n$ est appelé un *ordonnancement* : $\sigma(k) = 1$ signifie que le serveur k est actif, et $\sigma(k) = 0$ signifie que le serveur k est inactif. On note $q_t(k)$ le nombre de paquets en attente au serveur k. Par définition, chaque serveur actif à l'instant t transmet un paquet (s'il en a un à transmettre) dans l'intervalle de temps $[t, t+1)$, de telle sorte que si σ_t est l'ordonnancement choisi à l'instant t, alors on a la relation de récurrence suivante :
$$\mathbf{q}_{t+1} = (\mathbf{q}_t + \mathbf{a}_t - \sigma_t)^+.$$

En l'absence de contraintes, on choisirait $\sigma_t = \mathbf{1} = (1, \ldots, 1)$ afin de maximiser le nombre de paquets transmis (débit). Néanmoins, on souhaite précisément s'intéresser à des systèmes avec contraintes sur l'ensemble des serveurs qui peuvent être actifs en même temps. Ces contraintes seront codées par un ensemble $\mathcal{S} \subset \{0,1\}^n$ qui représente l'ensemble des ordonnancements possibles : ainsi, à chaque instant t, l'ordonnancement choisi σ_t doit faire partie de $\mathcal{S}$. Nous ferons une unique hypothèse sur la structure de $\mathcal{S}$: si σ est un ordonnancement admissible, alors tout sous-ordonnancement $\sigma' \leq \sigma$ est aussi admissible ($\sigma' \leq \sigma$ est à comprendre coordonnée par coordonnée : $\sigma'(k) \leq \sigma(k)$ pour tout $k = 1, \ldots, n$).

A l'instant t, on choisit un ordonnancement σ_t qui résout le problème d'optimisation suivant :
$$\sigma_t \in \arg\max_{\sigma \in \mathcal{S}} \sum_{k=1}^{n} \sigma_k w_k(\mathbf{q}_t(k))$$
où pour chaque k, on considère une fonction $w_k : \mathbb{R} \to [0, \infty)$. On rappelle qu'on a donc la relation de récurrence $\mathbf{q}_{t+1} = (\mathbf{q}_t + \mathbf{a}_t - \sigma_t)^+$.

Q1. On considère d'abord le cas $w_k(x) = x^2$, et on suppose que $E(a_1(k)^3) < +\infty$ pour tout $k = 1, \ldots, n$. Utiliser le critère de stabilité de Foster avec la fonction $V(\mathbf{q}) = \sum_{k=1}^n q(k)^3$ pour montrer que $(\mathbf{q}_t, t \geq 0)$ est positif récurrent lorsque λ est dans l'intérieur de l'enveloppe convexe de $\mathcal{S}$.

Q2. Montrer plus généralement que $(\mathbf{q}_t, t \geq 0)$ est positif récurrent lorsque λ est dans l'intérieur de l'enveloppe convexe de $\mathcal{S}$ si pour chaque $k \in \{1, \ldots, n\}$ les conditions suivantes sont satisfaites :

(1) w_k est croissante et continue ;

(2) $w_k(0) = 0$ et $\lim_{x \to +\infty} w_k(x) = +\infty$;

(3) pour tout $\delta > 0$, il existe $B_\delta \geq 0$ tel que pour tout $x \geq 0$,

$$E\left(w_k(x + a_0(k))a_0(k)\right) \leq (1 + \delta)w_k(x)\lambda_k + B_\delta$$

et

$$w_k((x-1)^+) \geq (1 - \delta)w_k(x) - B_\delta.$$

Q3. Le choix $w_k(x) = x$ garantit une région de stabilité maximale : quel intérêt peut-il donc y avoir à considérer des poids w_k ?