

## Contrôle sur le cours

### Modèles et Algorithmes d'Ingénierie de Trafic

On considère l'architecture d'une ferme de serveurs, telle que décrite sur la Figure 1. Dans cette architecture, on a  $K$  agents de routage qui reçoivent des requêtes (ou jobs) depuis l'extérieur et les routent vers des serveurs informatiques. On suppose que chaque agent de routage optimise indépendamment le temps de réponse de ses propres jobs. L'objectif est d'évaluer l'impact de la non-coordination des agents de routage sur les performances globales du système. Plus précisément, on veut comparer les performances obtenues avec un routage centralisé ( $K = 1$ ) et avec un routage décentralisé ( $K > 1$ ). Pour simplifier, si  $\lambda$  est le trafic total entrant dans le système (en nombre de jobs par seconde), on suppose que chaque agent de routage contrôle exactement la même quantité de trafic  $\frac{\lambda}{K}$ , et qu'il n'y a que deux serveurs.

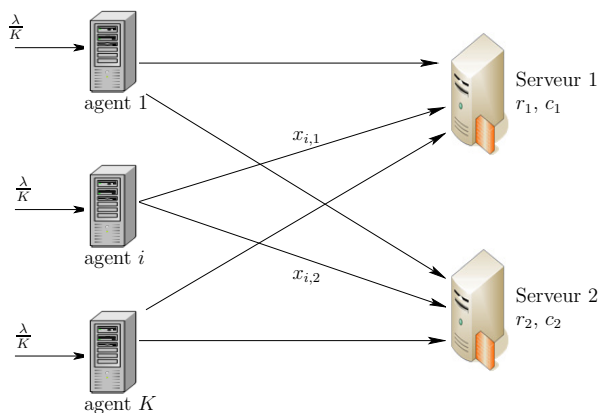


Figure 1: Architecture informatique considérée dans le sujet.

On note  $r_j$  la capacité (en job/s) du serveur  $j = 1, 2$  et on suppose que le coût de traitement d'un job sur ce serveur est  $c_j$  euros/s. On fait l'hypothèse que  $\frac{c_1}{r_1} < \frac{c_2}{r_2}$ , c'est-à-dire que le coût par job est inférieur sur le serveur 1. On note  $x_{i,j}$  la quantité de trafic envoyé par l'agent de routage  $i$  sur le serveur  $j$ , de sorte que le vecteur  $\vec{x}_i = (x_{i,1}, x_{i,2})$  représente la stratégie de routage de l'agent  $i$ . On a bien sûr  $x_{i,1} + x_{i,2} = \frac{\lambda}{K}$ . Le vecteur  $\vec{x} = (\vec{x}_1, \dots, \vec{x}_K) = (x_{1,1}, x_{1,2}, \dots, x_{K,1}, x_{K,2})$  représente lui la stratégie de routage utilisée par l'ensemble des agents de routage.

L'objectif de l'agent de routage  $i$  est de minimiser le temps de traitement moyen de ses propres jobs, c'est-à-dire de déterminer la stratégie de routage  $\vec{x}_i$  minimisant

$$T_i(\vec{x}_i, \vec{x}_{-i}) = c_1 \frac{x_{i,1}}{r_1 - y_1} + c_2 \frac{x_{i,2}}{r_2 - y_2}, \quad (1)$$

où  $y_j = \sum_{k=1}^K x_{k,j}$  représente le trafic total arrivant au serveur  $j = 1, 2$ . On notera que le routage optimal  $\vec{x}_i$  pour l'agent  $i$  dépend de la stratégie  $\vec{x}_{-i} = (\vec{x}_1, \dots, \vec{x}_{i-1}, \vec{x}_{i+1}, \dots, \vec{x}_K)$  utilisée par les autres agents  $k \neq i$ , dans la mesure où  $y_j = x_{i,j} + \sum_{k \neq i} x_{k,j}$ .

Pour évaluer la performance globale du système, on considère la métrique suivante

$$\begin{aligned} D(y_1, y_2) &= \sum_{i=1}^K T_i(\vec{x}_i, \vec{x}_{-i}), \\ &= c_1 \sum_{i=1}^K \frac{x_{i,1}}{r_1 - y_1} + c_2 \sum_{i=1}^K \frac{x_{i,2}}{r_2 - y_2}, \\ &= c_1 \frac{y_1}{r_1 - y_1} + c_2 \frac{y_2}{r_2 - y_2}, \end{aligned} \quad (2)$$

qui représente<sup>1</sup> le coût de traitement moyen d'un job dans le système.

## PREMIERE PARTIE: ROUTAGE CENTRALISE

Dans cette partie, on étudie le routage centralisé optimal, dans le cas où  $K = 1$ . Dans ce cas,  $\vec{x} = (y_1, y_2)$  et  $T_1(\vec{x}) = D(y_1, y_2)$  est la fonction objectif optimisée par l'unique agent de routage :

$$(OPT) \left\{ \begin{array}{l} \text{Minimiser :} \\ \text{sous les contraintes :} \end{array} \right. \quad \begin{array}{l} D(y_1, y_2) = c_1 \frac{y_1}{r_1 - y_1} + c_2 \frac{y_2}{r_2 - y_2} \\ y_1 + y_2 = \lambda, \\ y_1 \geq 0, y_2 \geq 0 \end{array}$$

On notera dans la suite  $(y_1^*, y_2^*)$  la stratégie de routage centralisée.

### Questions :

Q1.1 Quelles sont les conditions d'optimalité du routage ?

Q1.2 Sous quelle condition sur  $\lambda$ , le routage centralisé n'utilise que le premier serveur (i.e.  $y_1^* = \lambda$ ) ?

<sup>1</sup>A un facteur  $\lambda$  près, d'après la loi de Little.

Q1.3 Si les deux serveurs sont utilisés, quelle est la solution de routage  $(y_1^*, y_2^*)$  optimale ?

Q1.4 Calculer la performance  $D(y_1^*, y_2^*)$  à l'optimal pour les valeurs suivantes:  $r_2 = c_2 = 1, r_1 = 2, c_1 = 0.995, \lambda = 1$ .

## SECONDE PARTIE: ROUTAGE DECENTRALISE

Dans cette partie, on étudie le routage décentralisé, dans la cas où  $K > 1$ . On a vu avec l'équation (1) que la stratégie optimale pour l'agent  $i$  dépend de la stratégie  $\vec{x}_{-i}$  des autres joueurs. Cette situation correspond à ce qu'on appelle un jeu de routage non coopératif entre les agents de routage. L'objectif dans cette partie est d'étudier la performance globale du système à l'équilibre de Nash de ce jeu<sup>2</sup>. Une stratégie de routage  $\vec{x}^{ne}$  est un équilibre de Nash si aucun joueur ne peut améliorer ses performances par une déviation unilatérale du point  $\vec{x}^{ne}$ , c'est-à-dire si

$$\vec{x}_i^{ne} = \operatorname{argmin}_{\vec{x}_i} T_i(\vec{x}_i, \vec{x}_{-i}^{ne}),$$

pour tout joueur  $i = 1, \dots, K$ . On notera  $y_j^{ne}$  le trafic sur le serveur  $j$  à l'équilibre de Nash. On admettra dans un premier temps qu'à l'équilibre de Nash, tous les agents de routage utilisent la même stratégie :  $x_{i,j}^{ne} = \frac{y_j^{ne}}{K}$  pour tout  $i = 1, \dots, K$ .

### Questions :

Q2.1 Justifier succinctement que  $D(y_1^*, y_2^*) \leq D(y_1^{ne}, y_2^{ne})$ , ce qui explique qu'on puisse parler de dégradation des performances.

Q2.2 Montrer que si la stratégie de routage  $\vec{x}$  est telle que  $x_{i,j} = y_j/K$  pour  $j = 1, 2$  et  $i = 1, \dots, K$ , alors le coût marginal du serveur  $j$  pour le joueur  $i$  s'écrit sous la forme

$$\frac{\partial T_i}{\partial x_{i,j}}(\vec{x}_i, \vec{x}_{-i}) = \frac{c_j}{K} \frac{1}{r_j - y_j} \left( \frac{r_j}{r_j - y_j} + K - 1 \right).$$

Q2.3 Montrer que le vecteur  $(y_1^{ne}, y_2^{ne})$  est la solution optimale du problème suivant:

$$(P) \begin{cases} \text{Minimiser :} & F(y_1, y_2) = \sum_{j=1}^2 \frac{c_j}{K} \left[ \frac{y_j}{r_j - y_j} + (K - 1) \log \left( \frac{r_j}{r_j - y_j} \right) \right] \\ \text{sous les contraintes :} & y_1 + y_2 = \lambda, \\ & 0 \leq y_j < r_j, \quad j = 1, 2. \end{cases}$$

<sup>2</sup>On admettra ici que cet équilibre existe et qu'il est unique.

- Q2.4 En utilisant les valeurs de la question Q1.4 et  $K = 100$ , appliquez une itération de l'algorithme *Flow Deviation* au problème précédent en partant de la solution initiale  $y_1^{(0)} = y_2^{(0)} = \frac{1}{2}$ . On prendra le pas  $\beta = 0,996$ . Calculez les coûts marginaux  $\frac{\partial F}{\partial y_j}(y_1, y_2)$  des serveurs pour le point obtenu.
- Q2.5 En supposant que l'algorithme a convergé, calculez  $D(y_1^{ne}, y_2^{ne})$ . Quelle est la valeur du ratio  $\frac{D(y_1^{ne}, y_2^{ne})}{D(y_1^*, y_2^*)}$  ?
- Q2.6 **Question Bonus** : Montrer qu'à l'équilibre de Nash tous les agents utilisent la même stratégie de routage, c'est-à-dire  $x_{i,j}^{ne} = \frac{y_j^{ne}}{K}$ . *Indication* : on pourra tout d'abord montrer que  $x_{i,j}^{ne} < x_{k,j}^{ne}$  si et seulement si  $\frac{\partial T_i}{\partial x_{i,j}}(\vec{x}^{ne}) < \frac{\partial T_k}{\partial x_{k,j}}(\vec{x}^{ne})$ . On raisonnera ensuite par l'absurde pour montrer que, le routage des agents étant optimal,  $x_{i,1}^{ne} < x_{k,1}^{ne}$  conduit à une contradiction avec la conservation du trafic.