Equilibrage de Nash dans le problème d'allocation de tâches

Mostapha Zbakh

Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes, Rabat, Maroc

zbakh@ensias.ma

Résumé : Nous présentons le problème d'allocation de tâches comme un jeu non coopératif entre plusieurs joueurs. Nous donnons par la suite la structure de l'équilibre de Nash de ce jeu ainsi qu'un algorithme distribué qui permet de le trouver.

Mots-clés : Allocation de tâches, théorie des jeux, jeu non coopératif, équilibrage de Nash, algorithme distribué.

---

1. Problème d'Allocation de tâches comme un jeu non coopératif

La formulation générale d'un tel problème est la suivante. On suppose qu'on veuille traiter n tâches (égoïstes) de tailles T_1, T_2,…, T_n sur m machines quelconques m_i (i=1,…,m) de vitesses v_1,v_2,…,v_m. Définissons la charge L_i d'une machine i comme la somme des temps d'exécution des tâches qu'elle les traite et le coût d'une tâche comme la charge de la machine l'hébergeant. Le problème général d'allocation de tâches est de trouver une allocation de ces tâches pour que le coût total d'exécution soit minimal. Définissons le paramètre s_{ji} comme un réel compris entre 0 et 1 représentant la portion de la tâche j traitée par la machine i en supposant que les tâches sont divisibles. Nous appelons le vecteur $s_j=(s_{j1},s_{j2},…s_{jm})$ la stratégie d'allocation de la tâche j(j=1,…,n) et le vecteur s = (s_1,s_2, …, s_n) le profil de stratégie de ce jeu d'allocation de tâches.

Nous supposons que chaque machine est modélisée comme un système de fil d'attente M/M/1 (les arrivés suivent la loi de Poisson et le temps de traitement est Exponentiel). Le temps de réponse de la machine i est donné comme: $t_i(s)=\frac{1}{v_i-\sum_{j=1}^{n} s_{ji}T_j}$. Le coût d'une tâche j est donné donc par: $c_j(s)=\sum_{i=1}^{m} s_{ji}t_i(s)=\sum_{i=1}^{m}\frac{s_{ji}}{v_i-\sum_{j=1}^{n} s_{ji}T_j}$.

Notre but est de trouver une politique d'allocation de tâches faisable (s_1,s_2,…$s_{j,…,}$ s_n) [2] qui minimise les fonctions $c_j(s)$. La décision de chaque tâche j dépend des décisions des autres tâches puisque c_j est une fonction de s.

Définition 1: Le jeu non coopératif d'allocation de tâches est constitué d'un ensemble de joueurs, un ensemble de stratégies et des préférences entre les profils de stratégies. Les joueurs sont les n tâches et chaque tâche T_j a son ensemble de stratégies faisables d'allocation de tâches s_j. Chaque tâche j préfère le profil s que le profil s' si et seulement si $c_j(s) < c_j(s')$.

La solution de ce problème consiste donc à trouver l'équilibre de Nash[1, 2] de ce jeu d'allocation.

Définition 2[1,2,3]: L'équilibre de Nash de ce jeu d'allocation de tâches est un profil de stratégies s tel que pour chaque tâche j(j=1…n): s_j est tel que $c_j(s_1,s_2,...,s_j,...,s_n)=\min_{s^{\wedge}_j} c_j(s_1,s_2,...,s^{\wedge}_j,...s_n)$.

Pour ce jeu il existe un équilibre unique de Nash parce que les fonctions $c_j(s)$ sont continues, convexes et croissantes [4].

Afin de déterminer une solution pour notre jeu nous considérons une définition alternative de l'équilibre de Nash [3]: «l'équilibre de Nash peut être défini comme étant le profil de stratégies pour lequel la stratégie d'allocation de chaque tâche est une meilleure réponse aux stratégies des autres tâches ». La meilleure réponse d'une tâche fournira un temps de réponse minimal, en supposant que les stratégies des autres tâches sont maintenues fixes.

D'abord, nous déterminons les stratégies des meilleures réponses s_j pour chaque tâche j, après nous trouverons un profil de stratégies s = (s_1, s_2,…, s_n) pour lequel s_j est la meilleure réponse de la tâche j, pour j=1,2,…, n.

Soit $v_i^j = v_i - \sum_{k=1,k\neq j}^{m} s_{ki}T_k$ le taux de traitement disponible du processeur i comme il est vu par la tâche j. Le problème de calcul de la meilleure stratégie de réponse de la tâche j (j=1,…, n) se réduit au problème d'allocation d'une seule tâche sur les m machines ayant v_i^j comme taux de traitement, c'est-à-dire le calcul de la stratégie d'allocation optimale de cette tâche. Ceci peut être traduit en problème d'optimisation suivant (MEILLEURE-REPONSE$_j$):

$$\begin{cases} \min_{s_j} c_j(s) \text{ sous contraintes:} \\ s_{ji} \in [0,1],\ i=1,\ldots,m; \\ \sum_{i=1}^{m} s_{ji}=1\ ;\ \sum_{k=1}^{n} s_{ki}T_k \prec v_i^j, i=1,...,m\ ; \end{cases}$$

Théorème. Supposant que les machines sont classées par ordre décroissant de leurs taux de traitement disponible $\left(v_1^j \geq v_2^j \geq ... \geq v_n^j\right)$, la solution s_j du problème d'optimisation **MEILLEURE-REPONSE$_j$** est donnée par:

$s_{ji} = \begin{cases} \frac{1}{T_j}\left(v_i^j - \sqrt{v_i^j}\,\frac{\sum_{i=1}^{c_j} v_i^j - T_j}{\sum_{i=1}^{c_j}\sqrt{v_i^j}} \right) & \text{si } 1 \le i < c_j \text{ où } c_j \text{ est l'indice minimal qui vérifie l'inégalité } \sqrt{v_{c_j}^j} \le \frac{\sum_{k=1}^{c_j} v_k^j - T_j}{\sum_{k=1}^{c_j}\sqrt{v_k^j}} \\ 0 \text{ si } c_j \le i < \le m \end{cases}$

2. Un algorithme distribué d'allocation de tâches

En se basant sur les travaux effectués dans [3], nous décrivons un algorithme distribué pour calculer l'équilibre de Nash. Pour cela, on procède par une généralisation de ce problème de la façon suivante. Au lieu de parler d'une tâche j, on parlera d'une source de génération des tâches j. Ainsi la source j produira des tâches identiques et de même taille T_j. L'idée de l'algorithme est la suivante. Les sources génèrent les tâches en parallèle pendant plusieurs itérations. Dans chaque itération, on mesure la norme L^1 comme $\sum_{j=1}^{m}\left|c_j^{(l-1)} - c_j^l\right|$. On arrête lorsqu'on obtient une valeur inférieure à un seuil d'erreur prédéfini.

Le calcul de cet équilibre de Nash peut exiger une certaine coordination entre les sources (afin d'obtenir l'information sur la charge de chaque machine). Nous utilisons les notations suivantes en plus de ceux de la section précédente :

$j \leftarrow$ *le numéro de la source j ;*

$l \leftarrow$ *le numéro de l'itération ;*

$s_j^{(l)} \leftarrow$ *la stratégie de j calculée à l'itération l ;*

$c_j^{(l)} \leftarrow$ *le temps d'exécution de j à l'itération l ;*

$\varepsilon \leftarrow$ *le seuil d'erreur ;*

$norme \leftarrow$ *la norme* L^1 *à l'itération l, définie comme* $\sum_{j=1}^{n}\left|c_j^{(l-1)} - c_j^l\right|$;

$envoyer(j, mesg) \leftarrow$ *envoie le message mesg à la source j ;*

$recevoir(j, mesg) \leftarrow$ reçoit le message *mesg de la source j ;*

Chaque source j exécute l'algorithme suivant :

1. Initialisation :

$s_j^{(0)} \leftarrow 0;$

$c_j^{(0)} \leftarrow 0;$

$l \leftarrow 0;$

$norme \leftarrow 1;$

$somme \leftarrow 0;$

$etat \leftarrow CONTINUE;$

$gauche \leftarrow [(j-2) \bmod n] + 1;$

$droite \leftarrow [j \bmod n] + 1;$

2.Tant que (1) faire

si *(j=1) {source 1}*

si *(*$l \neq 0$*)* ***recevoir****(gauche,(norme,l,etat)) ;*

si *(*$norme \prec \varepsilon$*)* ***envoyer****(droite,(norme,l,STOP)) ;*

exit

$somme \leftarrow 0;$

$l \leftarrow l + 1;$

else *{autres sources}*

recevoir*(gauche,(somme,l,etat)) ;*

si *(etat=STOP)*

si $(j \neq n)$ ***recevoir****(droite,(somme,l,STOP)) ;*

Pour $i := 1,\ldots,m$ ***faire***

Obtenir v_i^j *en examinant la file d'attente de chaque machine*

$\left(v_i^j \leftarrow v_i - \sum_{k=1, k\neq j}^{m} s_k T_k\right)$

$s_j^{(l)} \leftarrow \text{MEILLEURE_REPONSE}_j\left(v_1^j,\ldots,v_m^j, T_j\right);$

Calcule de $c_j^{(l)}$;

$somme \leftarrow somme + \left|c_j^{(l-1)} - c_j^{(l)}\right|$;

envoyer*(droite,(somme,l,CONTINUE)) ;*

fintantque

3. Conclusion et perspective

Nous avons formulé le problème général d'allocation de tâches comme un jeu non coopératif entre plusieurs joueurs. Nous avons proposé ensuite la structure de l'équilibre de Nash de ce jeu. Basons sur cette structure, nous avons décrit un algorithme distribué pour le trouver. Plusieurs adaptations et extensions possibles pour ce jeu d'équilibrage, notamment dans les grilles informatiques et Internet.

4. Bibliographie

1. M. Osborne, "An Introduction to Game Theory", Oxford University Press, New York, 2004.
2. J. Nash, "Non-cooperative games", Ann. Math. 54 (2) 286-295, 1951
3. D. Grosu, A.T. Chronopoulos, "Noncooperative load balancing in distributed systems", J. of Parallel Distrib. Comput. 65, 1022-1034, 2005
4. A. Orda, R. Rom, N. Shimkin, "Competitive routing in multiuser communication networks", IEEE/ACM Trans. Networking 1 (5), 510-521, 1993