Optimisation du préchargement dans un monde dynamique

Kausthub Keshava[1], Alain Jean-Marie[2], Sara Alouf[3]

[1] Deloitte India (Offices of the US), Hyderabad, 500032 Hyderabad, Inde
kkeshava@deloitte.com

[2] Inria, Université de Montpellier, 34095 Montpellier, France
Alain.Jean-Marie@inria.fr

[3] Inria, Université Côte d'Azur, 06902 Sophia Antipolis, France
Sara.Alouf@inria.fr

Mots-clés : *préchargement, optimisation, processus de décision markovien, arbres aléatoires*

1 Introduction

Le préchargement (en Anglais, *prefetching*) est une technique classique en informatique. Son principe est de réduire le temps d'accès à un « objet » en allant le chercher avant d'en avoir effectivement besoin. Cela implique fondamentalement un compromis entre le coût d'utiliser les ressources (réseau, mémoire) et le coût de la latence pour accéder à cet objet s'il n'a pas été préchargé. Dans certains cas, comme pour la navigation sur le Web, le « contrôleur » chargé de gérer la mémoire ne connaît pas à l'avance l'ordre dans lequel le « surfeur » va visionner les hyperdocuments. Une modélisation stochastique est alors appropriée pour étudier le compromis en question, et le formalisme adéquat est celui des Processus de Décision Markoviens (MDPs).

2 Le modèle

Le surfeur aléatoire. Le principe d'utiliser les MDPs pour optimiser le préchargement dans un contexte de navigation vidéo, a été proposé dans [1, 3] et plusieurs généralisations existent. Ces modèles sont appliquables à d'autres types de documents. Leur point commun est que l'information accessible est représentée par un graphe orienté dont les nœuds sont les documents et les arcs les transitions possibles entre documents. Le surfeur est supposé aller de document en document en suivant ces liens, de façon probabiliste. Le contrôleur utilise le temps entre deux mouvements pour précharger des documents. Nous considérons ici une version simple dans laquelle le contrôleur a la capacité de précharger un nombre entier et constant k de documents entre deux mouvements du surfeur. La question générique est de décider quels document précharger afin de minimiser la probabilité que le surfeur accède à un document non présent.

La situation étudiée jusqu'ici est celle où le graphe entier des documents est connu du contrôleur. Les méthodes numériques standard pour les MDPs permettent de calculer le contrôle optimal, mais elles sont vite limitées par la taille exponentielle de l'espace de décision et de l'espace d'états. Le sous-problème de savoir s'il existe une politique de préchargement parfaite (le surfeur est toujours satisfait) est étudié dans [2] et la conclusion est que le problème est difficile, sauf dans les arbres où la question peut être décidée en temps polynomial. En dehors de cette situation de coût nul dans un arbre, aucun cas n'est connu où une politique de structure simple est optimale. Cette situation d'information complète sur l'espace des documents ne procure donc pas d'algorithme de décision opérationnel.

Le graphe de documents aléatoire. Nous considérons donc ici que le contrôleur ne connaît qu'une partie du graphe de documents, qu'il découvre au fur et à mesure des mouvements du surfeur. Cela permet de réduire l'espace de décision, mais également le stockage du graphe, et cela peut correspondre à certaines situations pratiques. L'hypothèse la plus simple est que le graphe connu est un arbre de profondeur donnée d et l'évolution du système est comme suit. Le surfeur se trouve à la racine. Le contrôleur peut « marquer » k nœuds de l'arbre : les documents correspondants sont supposés préchargés. Le surfeur se déplace ensuite au hasard, uniformément, sur l'un des fils de la racine. Le surfer ne revient jamais en arrière et le reste de l'arbre est oublié. Si ce fils n'est pas marqué, cela coûte une unité au contrôleur. Les feuilles sont alors complétées par une nouvelle génération de feuilles. On suppose leur nombre choisi aléatoirement uniformément entre 1 et un nombre p. Ce modèle a donc trois paramètres : k, d et p. Il est supposé que $k < p$ afin que le problème ne soit pas trivial.

3 Résultats

Nous avons déterminé la politique optimale pour certaines valeurs des paramètres, dans le cas où le critère d'optimisation est l'espérance du coût total avec horizon fini, et dans le cas du coût moyen en horizon infini. Les premiers résultats consacrent les politiques qui sont « gloutonnes » dans le sens qu'elles marquent en premier les fils de la racine.

Théorème 1 *Si $k = 1$, il est optimal de marquer n'importe quel fils de la racine. Si $d = 1$, il est optimal de marquer n'importe quels k fils de la racine.*

Comme conséquence, le coût moyen optimal pour $d = 1$ ou $k = 1$ peut être calculé comme :

$$J^* = \frac{1}{p} \sum_{m=k+1}^{p} \frac{m-k}{m} = 1 - \frac{k}{p} (1 + \mathbb{H}_p - \mathbb{H}_k)$$

où $\mathbb{H}_p$ est le p-ième nombre harmonique. Cette valeur est une borne supérieure sur le coût moyen optimal, en général.

Pour les arbres de profondeur $d = 2$ et un budget de préchargement $k = 2$, nous définissons une politique nommée « Greedy Finite Optimal » (GFO). Si plus de deux fils sont non marqués, elle en marque deux. Si un seul fils est non marqué, il est marqué et la second marque va dans le plus petit des sous-arbres dont la taille est ≥ 3, ou le plus grand s'il n'y en a aucun. Si tous les fils sont marqués, ce qui ne peut se produire que s'il y a un ou deux fils, les deux marques vont dans le plus petit sous-arbre s'il est de taille ≥ 4, ou sont réparties entre les deux sous-arbres sinon. On montre alors le résultat suivant.

Théorème 2 *Si $d = 2$, $k = 2$ et $p \leq 5$, la politique GFO est optimale pour le coût moyen.*

Le formalisme et les détails sont présentés dans [4]. Les preuves sont basées sur l'équation d'optimalité de Bellman spécifique au critère. Dans le cas de l'horizon fini, une récurrence sur la longueur de l'horizon est utilisée. Dans le cas du coût moyen, la solution de l'équation d'optimalité est exhibée. Les recherches futures se focaliseront sur la politique Greedy Finite Optimal : est-elle optimale pour $p \geq 6$, comment se généralise-t-elle pour $k \geq 3$, et quelle est sa performance avec d'autres graphes que les arbres ?

Références

[1] V. Charvillat and R. Grigoraş. Reinforcement learning for dynamic multimedia adaptation. *Journal of Network and Computer Applications*, 30(3) :1034–1058, 2007.

[2] F. Fomin, F. Giroire, A. Jean-Marie, D. Mazauric, and N. Nisse. To satisfy impatient web surfers is hard. *Journal of Theoretical Computer Science*, 526(20) :1–17, March 2014.

[3] R. Grigoraş, V. Charvillat, and M. Douze. Optimizing hypervideo navigation using a Markov decision process approach. In *ACM Multimedia*, pages 39–48, 2002.

[4] K. Keshava, A. Jean-Marie, and S. Alouf. Optimal prefetching in random trees. *Mathematics*, 9(19) :2437, Oct 2021.