

Optimisation en temps-réel du téléchargement de vidéos

Olivia Morad², Alain Jean-Marie^{1,2} *

¹ INRIA

² LIRMM, CNRS et Université de Montpellier 2, 161 Rue Ada, 34392 Montpellier Cedex 5
{Olivia.Morad,ajm}@lirmm.fr

Mots-Clés : *Vidéo à la demande, Markov Decision Process*

1 Introduction

La Vidéo (ou télévision) à la Demande se présente comme un des services majeurs rendus possibles par l'augmentation des performances des réseaux et de l'informatique domestique ou portable. Ce type de service permet aux utilisateurs, en théorie, de choisir de leur domicile un document vidéo dans une grande collection, et le visionner comme s'il utilisait un lecteur de vidéos ou de DVD conventionnel [1].

Du point de vue de l'ordinateur qui exécute l'application, le problème principal est de s'assurer que les parties de la vidéo sont récupérées du réseau à temps. Si le visionnage est séquentiel, cela ne nécessite pas de planification particulière : juste un débit suffisant (le cas du *streaming* simple). Mais si l'utilisateur décide de sauter d'un endroit à l'autre du document, cela devient problématique. Dans les applications multimédia futures, comme celle développée dans le projet ANR VOODDO, la séquence vidéo ne sera plus nécessairement « linéaire » et l'utilisateur interagira constamment avec : cela va nécessiter la construction de *plans de téléchargement* sophistiqués et dynamiques, qui diront quelle information il est utile de récupérer du réseau à chaque instant.

Cet article rend compte d'un premier pas vers la conception de tels algorithmes. Nous proposons un modèle simplifié de l'interaction utilisateur/réseau dans un contexte de décision temps-réel. Ce modèle servira de base pour la construction et le test d'algorithmes et d'heuristiques.

2 Le Modèle

Ce problème d'optimisation du plan de téléchargement peut être vu comme un problème de décision séquentielle : le contrôleur observe l'état du système, et choisit une action. Cela induit un coût ou une récompense, et influence l'évolution du système jusqu'à un nouvel état. On recommence alors la séquence observation/action. Nous avons adopté un des formalismes pour de tels problèmes : les *Markov Decision Processes (MDP)* [2], et nous avons considéré un scénario simplifié mais représentatif. Notre hypothèse est que la vidéo est segmentée en unités appelées « blocs ». L'utilisateur choisit

*Ce travail est financé par l'action VOODDO du programme « Audiovisuel & Multimédia » de l'ANR.

indépendamment du système les blocs qu'il souhaite voir et dans quel ordre, mais il regarde ces blocs entièrement. Les blocs sont téléchargés du réseau et stockés. Un blocage intervient quand l'utilisateur souhaite voir un bloc qui n'est pas entièrement présent. Dans ce cas, le contrôleur consacre toute la bande passante du réseau à récupérer la partie manquante et l'utilisateur ne peut qu'attendre. Sinon, le contrôleur est libre de télécharger en parallèle les blocs qu'il souhaite pendant que l'utilisateur visionne le bloc choisi. La bande passante du réseau est partagée également entre ces téléchargements (pas de priorité). Nous avons modélisé ce système comme un MDP avec les éléments suivants :

Instants de décision : le temps est discret : le contrôleur prend ses décisions aux instants où l'utilisateur commence ou finit de visionner un bloc. L'horizon est infini *a priori* mais voir plus bas.

États : ils sont constitués du numéro du bloc en cours de visionnage et, pour chaque bloc, de la proportion de ce bloc actuellement en mémoire. L'espace d'état obtenu est fini si les proportions sont restreintes dans un ensemble fini, par exemple en discrétisant l'intervalle $[0, 1]$.

Actions : ce sont les sous-ensembles de $\{1, 2, \dots, B\}$, où B est le nombre total de blocs.

Probabilités de transition : l'utilisateur passe du bloc b à b' suivant une probabilité de transition $p_{bb'}$ connue. Les transitions sur les proportions sont déterministes : elles dépendent du nombre de téléchargements en parallèle, du débit du réseau, et de la durée nécessaire pour visionner un bloc.

Coût et critère d'optimalité : quand le bloc choisi n'est pas en mémoire, cela induit un coût instantané qui dépend du temps nécessaire pour récupérer l'information manquante au débit maximal. Pour l'optimisation globale, nous avons choisi le coût total sans actualisation. En effet, bien que l'horizon soit infini, il existe un ensemble d'états *terminal* (par exemple, quand tous les blocs sont en mémoire) qui n'induit aucun coût. Ce problème est donc dans la catégorie des *stochastic shortest path* [3] ou des *negative models* [2].

3 Calcul de la décision optimale

Résoudre le problème consiste en le calcul d'une règle de rétroaction : état \mapsto action, qui peut être stationnaire d'après la théorie. Nous avons implémenté les algorithmes de MDP « standards » pour la calculer : *Value Iteration*, *Policy Iteration* et *Modified Policy Iteration* [2]. Il est bien connu que ces algorithmes sont limités par la « malédiction de la dimension ». Nous comptons nous servir d'eux sur des exemples petits, pour avoir un point de référence (solution exacte), déterminer des propriétés structurelles et guider la constructions d'heuristiques.

Travail futur. La solution de problèmes résultant de taille réaliste et/ou de modèles plus complets, va nécessiter l'utilisation d'heuristiques d'Intelligence Artificielle. Ainsi, nous prévoyons : a) de résoudre ce modèle simplifié en utilisant des algorithmes génétiques et des réseaux neuronaux ; b) de construire des modèles étendus, complètement puis partiellement observés, et les résoudre sur la base d'algorithmes d'apprentissage ; c) valider le tout en utilisant un simulateur.

Références

- [1] D. Ghose and H. Kim, Scheduling video streams in video-on-demand systems : a survey, *Multimedia Tools and Applications*, 11 (2), 167–195, 2000.
- [2] M.L. Puterman, *Markov Decision Processes*, John Wiley & Sons, Hoboken, New Jersey, 2005.
- [3] D.P. Bertsekas, *Dynamic programming and Optimal Control*, Athena Scientific, USA, 2000.