Minimiser le temps d'attente pour un service de navettes

Laurent Daudet[1], Frédéric Meunier[1]

CERMICS, École des Ponts ParisTech
6 et 8 avenue Blaise Pascal Cité Descartes - Champs-sur-Marne 77455 Marne-la-Vallée Cedex 2
laurent.daudet@cermics.enpc.fr, frederic.meunier@enpc.fr

Mots-clés : *Algorithme polynomial, algorithme approché, schéma d'approximation, plus court chemin, transport ferroviaire*

1 Contexte

Considérons un terminal de chargement sur lequel arrivent continûment des clients et une flotte de navettes pour les transporter. Comment planifier les départs des navettes afin de minimiser le temps d'attente des clients ? Ce problème est inspiré d'une étude réalisée par les auteurs dans le cadre d'un partenariat avec Eurotunnel. Une des problématiques principales d'Eurotunnel est la congestion sur le terminal de chargement, notamment due aux files d'attente de clients attendant leur départ. Ce problème se situe à l'interface des files d'attente (étudiées par exemple dans [4]) et de la planification ferroviaire [2, 5]. Ce type de problème apparaît aussi dans d'autres domaines que le transport. Dans le protocole TCP, un serveur est confronté à un flux entrant de paquets, qu'il doit traiter par "batch" tout en cherchant à minimiser le temps d'attente des paquets (voir par exemple les travaux de Dooly, Goldman et Scott [3]). Un autre exemple dans le même esprit a été étudié par Barbosa et Friedman [1] : ils considèrent un modèle de lot-sizing dynamique, avec une demande continue variable, qui doit être satisfaite par des réapprovisionnements ponctuels. On peut aussi imaginer une application dans l'industrie chimique : un test doit être être réalisé sur des produits chimiques arrivant continûment sur une ligne. Le test prend un certain temps et peut être réalisé sur plusieurs produits simultanément. Quel est le planning de tests minimisant le temps total ? Plus généralement, la question posée dans cet abstract est pertinente dès que l'on a à traiter par lot une demande variable, continue, connue à l'avance et dont on veut minimiser le temps d'attente avant traitement.

2 Présentation des problèmes

On considère le problème sur une journée, modélisée comme un intervalle $[0, T]$. On se donne une flotte de S navettes dont on veut déterminer les horaires de départ. Ces départs doivent être calculés en amont, quelques jours avant la journée considérée. Il s'agit donc de la planification offline d'une grille horaire des départs des navettes. Celles-ci sont toutes de capacité $C \geq 0$ et situées sur le même terminal. Les clients sont infinitésimaux et arrivent continûment sur le terminal. Ces arrivées sont décrites par une fonction $D : [0, T] \to \mathbb{R}_+$ qui à un instant t associe le nombre total de clients arrivés sur l'intervalle $[0, t]$. On suppose que D est semi-continue supérieurement.

Lorsque les horaires de départs sont fixés, le processus de chargement est le suivant : chaque client arrivant sur le terminal est placé dans une file d'attente. Les clients ne sont chargés directement dans une navette qu'une fois que tous les clients pouvant être transportés dans cette navette sont présents, sans mettre en retard le départ de celle-ci. Charger une navette avec x clients prend un temps νx, avec $\nu \geq 0$, et la navette part dès que son chargement est terminé. Le cas où ν est nul modélise le problème où les clients ne passent pas par les files d'attente et sont chargés dans la navette sans attendre le dernier client. La politique First In

First Out qui assure qu'un client arrivant avant un autre ne parte pas strictement après ce dernier doit de plus être respectée pour les départs. Deux versions sont considérées concernant les navettes. Dans une première, le retour des navettes n'est pas autorisé : une navette quittant le terminal ne peut pas être réutilisée pour une autre mission. Dans une seconde, le retour est permis : une navette quittant le terminal revient après une durée de trajet $\pi \geq 0$ et est disponible à nouveau pour une autre mission. Pour chacune des deux versions, deux critères sur le temps d'attente sont considérés : le temps d'attente maximal sur tous les clients et le temps d'attente moyenné sur tous les clients.

On a donc quatre problèmes : lorsque le retour n'est pas autorisé, on a respectivement pour les critères du temps d'attente maximal et moyen les problèmes $\mathrm{P}^{\max}_{\text{no return}}$ et $\mathrm{P}^{\text{ave}}_{\text{no return}}$. De même lorsque le retour est autorisé, on a $\mathrm{P}^{\max}_{\text{return}}$ et $\mathrm{P}^{\text{ave}}_{\text{return}}$.

3 Modèles

Pour modéliser ces problèmes, nous avons besoin d'introduire les pseudo-inverses de D définies par $\tau\colon y \mapsto \inf\{t \in \mathbb{R}_+\colon D(t) > y\}$ et $\bar{\tau}\colon y \mapsto \inf\{t \in \mathbb{R}_+\colon D(t) \geq y\}$.

Une solution est identifiée par une séquence $\boldsymbol{y} = (y_j)$ croissante, où y_j modélise le nombre total de clients ayant quitté le terminal dans les j premières missions, et une séquence $\boldsymbol{\ell} = (\ell_j)$ représentant les instants de début de chargement pour chaque départ. Les missions sont numérotées suivant l'ordre des départs. On donne le départ la mission j comme étant $\ell_j + \nu(y_j - y_{j-1})$, c'est à dire immédiatement après l'instant où le dernier client est chargé dans la navette, quitte à retarder un peu l'instant de début de chargement pour respecter la politique First In First Out. Pour les problèmes $\mathrm{P}^{\max}_{\text{no return}}$ et $\mathrm{P}^{\text{ave}}_{\text{no return}}$, les solutions sont de taille S, alors que pour les problèmes $\mathrm{P}^{\max}_{\text{return}}$ et $\mathrm{P}^{\text{ave}}_{\text{return}}$, elles sont de taille infinie car on ne connaît pas *a priori* le nombre de missions nécessaires à l'optimum.

Les valeurs des fonctions objectifs sont respectivement pour les problèmes de minimisation du temps d'attente maximal et du temps d'attente moyenné sont notées $g^{\max}(\boldsymbol{\ell}, \boldsymbol{y})$ et $g^{\text{ave}}(\boldsymbol{\ell}, \boldsymbol{y})$. On peut interpréter $\tau(y_{j-1})$ comme l'instant d'arrivée du premier client de la mission j.

$$\begin{aligned} g^{\max}(\boldsymbol{\ell}, \boldsymbol{y}) &= \max_j \left(\mathbf{1}_{y_j > y_{j-1}}(\ell_j + \nu(y_j - y_{j-1}) - \tau(y_{j-1}))\right), \\ g^{\text{ave}}(\boldsymbol{\ell}, \boldsymbol{y}) &= \frac{1}{D(T)} \left(\sum_j \int_{y_{j-1}}^{y_j} (\ell_j + \nu(y_j - y_{j-1}) - \bar{\tau}(y)) dy \right). \end{aligned}$$

Les problèmes $\mathrm{P}^{\max}_{\text{no return}}$ et $\mathrm{P}^{\text{ave}}_{\text{no return}}$ peuvent s'écrire sous la forme

$$\begin{array}{lllr} \text{Min} & g(\boldsymbol{\ell}, \boldsymbol{y}) & & \\ \text{s.t.} & y_j - y_{j-1} \leq C & j = 1, \ldots, S & \text{(i)} \\ & y_{j-1} \leq y_j & j = 1, \ldots, S & \text{(ii)} \\ & y_S = D(T) & & \text{(iii)} \\ & \bar{\tau}(y_j) \leq \ell_j & j = 1, \ldots, S & \text{(iv)} \\ & \ell_j + \nu(y_j - y_{j-1}) \leq \ell_{j+1} + \nu(y_{j+1} - y_j) & j = 1, \ldots, S-1 & \text{(v)} \\ & y_0 = 0, & & \end{array} \qquad (\mathrm{P}_{\text{no return}})$$

où g est soit $g^{\max}$, soit g^{ave}. La contrainte (i) assure que la capacité de la navette est respectée. La contrainte (ii) assure que les indices des variables y_j sont consistants. La contrainte (iii) assure que tous les clients ont quitté le terminal après les S missions. La contrainte (iv) assure que le chargement commence après l'arrivée du dernier client et la contrainte (v) assure la politique de départs pour les clients.

Les problèmes $\mathrm{P}^{\max}_{\text{return}}$ and $\mathrm{P}^{\text{ave}}_{\text{return}}$ peuvent eux s'écrire sous la forme,

$$\begin{array}{lllr}
\text{Min} & g(\boldsymbol{\ell}, \boldsymbol{y}) & & \\
\text{s.t.} & y_j - y_{j-1} \leq C & j \in \mathbb{N}^* & \text{(i)} \\
& y_{j-1} \leq y_j & j \in \mathbb{N}^* & \text{(ii)} \\
& \lim_{j\to+\infty} y_j = D(T) & & \text{(iii)} \\
& \bar{\tau}(y_j) \leq \ell_j & j \in \mathbb{N}^* & \text{(iv)} \\
& \ell_j + \nu(y_j - y_{j-1}) \leq \ell_{j+1} + \nu(y_{j+1} - y_j) & j \in \mathbb{N}^* & \text{(v)} \\
& \ell_j + \nu(y_j - y_{j-1}) + \pi \leq \ell_{j+S} & j \in \mathbb{N}^* & \text{(vi)} \\
& y_0 = 0, & &
\end{array} \qquad (\mathrm{P}_{\text{return}})$$

où g est soit $g^{\max}$, soit g^{ave}. Les contraintes (i), (ii), (iv) et (v) sont les mêmes que précédemment. La contrainte (iii) est aussi la même mais adaptée au fait qu'on a potentiellement un nombre infini de missions. On a ajouté la contrainte (vi) qui assure que le chargement commence après que la navette est revenue sur le terminal.

4 Algorithmes et résultats

On considère la demande D, la capacité C, le taux de chargement ν et la durée de la journée T comme des constantes du problème. Seule la taille de la flotte de navettes S peut varier.

On donne une idée d'algorithme pour le problème $\mathrm{P}^{\max}_{\text{no return}}$ approchant la valeur de l'optimum à ρ près, où $\rho > 0$. L'algorithme consiste en une dichotomie et permet de calculer très rapidement des solutions arbitrairement proches de la valeur de l'optimum.

On réalise une dichotomie sur h, la valeur optimale du problème. Tant que les bornes h^+ et h^- sont telles que $h^+ - h^- > \rho$, on considère un système $\mathcal{S}^h$ dont l'ensemble des solutions est vide si et seulement si l'ensemble des solutions réalisables de $\mathrm{P}^{\max}_{\text{no return}}$ ayant une valeur inférieure à h est vide. Décider si $\mathcal{S}^h$ est vide est polynomial. On met alors à jour les bornes h^+ et h^- de h jusqu'à arriver au critère d'arrêt.

Théorème 1. *Soit $\rho > 0$. L'algorithme ci-dessus calcule en $O\left(S \log \frac{1}{\rho}\right)$ une solution réalisable $\boldsymbol{y}$ du problème $\mathrm{P}^{\max}_{\text{no return}}$ de valeur $g^{\max}_{\text{no return}}(\boldsymbol{y}) \leq OPT + \rho$, où OPT est la valeur optimale du problème.*

La complexité cache ici les données D, C, ν et T prises comme constantes du problème, elle ne dépend que de la taille de la flotte S et de l'erreur ρ voulue avec la valeur optimale.

Dans le cas particulier où D est strictement croissante, le corollaire suivant prouve l'existence d'un algorithme polynomial calculant une $(1+\varepsilon)$-approximation de la valeur optimale du problème $\mathrm{P}^{\max}_{\text{no return}}$, i.e. une solution réalisable de valeur $g \leq (1+\varepsilon)OPT$, où OPT est la valeur optimale. On appelle ces algorithmes des schémas d'approximation polynomiaux.

Corollaire 1. *Soit $\varepsilon > 0$. Si D est strictement croissante, alors l'algorithme ci-dessus calcule en $O\left(S \log \frac{S}{\varepsilon}\right)$ une $(1+\varepsilon)$-approximation de la valeur optimale $\mathrm{P}^{\max}_{\text{no return}}$.*

La complexité énoncée ici est la complexité réelle du problème, ne dépendant pas des constantes fixées précédemment.

On donne maintenant une idée d'algorithme approchant la valeur optimale du problème $\mathrm{P}^{\text{ave}}_{\text{no return}}$. Pour une précision $\varepsilon > 0$, il s'agit de construire un graphe $G_\varepsilon = (\mathcal{V}_\varepsilon, \mathcal{A}_\varepsilon)$ sans circuit et de fournir une fonction de poids bien choisie w sur les arcs. On calcule alors un chemin p entre un sommet origine et un sommet destination minimisant $\sum_{a\in A(p)} w(a)$, où $A(p)$ désigne l'ensemble des arcs de p. Ce chemin fournit alors une solution $(\boldsymbol{\ell}, \boldsymbol{y})$ réalisable pour $\mathrm{P}^{\text{ave}}_{\text{no return}}$ et de valeur à ε de la valeur optimale.

Théorème 2. *Soit $\varepsilon > 0$. Supposons que D possède des dérivées à droite en tout point. Si $\inf_{t\in[0,T]} D'_+(t) > 0$, alors l'algorithme ci-dessus calcule en $O\left(\frac{S^6}{\varepsilon^5}\right)$ une $(1+\varepsilon)$-approximation de la valeur optimale de $\mathrm{P}^{\text{ave}}_{\text{no return}}$.*

Sous certaines hypothèses sur la demande et sur le chargement, il est même possible de donner un algorithme exacte pour les problèmes $\mathrm{P}^{\max}_{\text{return}}$ et $\mathrm{P}^{\text{ave}}_{\text{return}}$.

Proposition 1. *Supposons D constante par morceaux définies par K intervalles faisant partie de l'input, et $\nu = 0$. Pour chacun des problèmes $\mathrm{P}^{\max}_{\text{no return}}$ et $\mathrm{P}^{\text{ave}}_{\text{no return}}$, il existe un algorithme calculant en $O(K^2S^3)$ une solution optimale. Si $C = +\infty$, la complexité est réduite à $O(K^2S)$.*

La complexité énoncée ici est la complexité réelle du problème. L'hypothèse que D soit constante par morceaux signifie que les clients n'arrivent qu'à un nombre fini d'instants. Cela couvre notamment le cas où les clients ne sont plus infinitésimaux. L'hypothèse $\nu = 0$ signifie que les clients ne passent pas par les files d'attente et sont chargés dans la navette sans attendre le dernier client.

De la même manière que pour les problèmes sans retour, on peut donner des algorithmes d'approximation pour les problèmes $\mathrm{P}^{\max}_{\text{return}}$ et $\mathrm{P}^{\text{ave}}_{\text{return}}$. On fixe comme constantes les mêmes données que pour le problème sans retour, en ajoutant la taille de la flotte de navettes S. La complexité de ces algorithmes est ici relative à la taille de la sortie notée N.

Théorème 3. *Soit $\varepsilon > 0$. Supposons que D possède des dérivées à droite en tout point. Si $\inf_{t\in[0,T]} D'_+(t) > 0$, alors pour chacun des problèmes $\mathrm{P}^{\max}_{\text{return}}$ et $\mathrm{P}^{\text{ave}}_{\text{return}}$, il existe un algorithme polynomial calculant en $O\left(\frac{N^{5S+1}}{\varepsilon^{4S+1}}\right)$ une $(1+\varepsilon)$-approximation de la valeur optimale.*

Nous avons traité un dernier cas particulier du problème avec retour des navettes. Il s'agit du cas où tous les clients sont présents au début de la période, i.e. pour tout $t \in [0, T]$, $D(t) = D(T)$. Nous donnons ici un résultat lorsque la flotte n'est composée que d'une seule navette. Le cas du problème $\mathrm{P}^{\max}_{\text{return}}$ est assez simple et l'on a une forme close d'une solution optimale. Pour le problème $\mathrm{P}^{\text{ave}}_{\text{return}}$, il existe un algorithme efficace pour en calculer une dès que $\pi > 0$.

Proposition 2. *Supposons $S = 1$ et que $D(t) = D(T)$ pour tout $t \in [0, T]$. Si $\pi > 0$, la valeur optimale de $\mathrm{P}^{\text{ave}}_{\text{return}}$ peut être calculée en $O\left(\frac{D(T)}{C}\right)$ et une solution optimale en $O\left(\frac{\nu C}{\pi} + \frac{D(T)}{C} + 1\right)$.*

Le cas $\pi = 0$ est aussi intéressant à étudier car il n'y a pas de solution optimale, mais $\mathrm{P}^{\text{ave}}_{\text{return}}$ possède une valeur optimale.

Une étape délicate de la preuve de la proposition consiste à montrer que $\mathrm{P}^{\text{ave}}_{\text{return}}$ a une solution optimale à support fini dans le cas où $\pi > 0$. Ce résultat est utilisé dans la construction des algorithmes du Théorème 3 pour chacun des problèmes $\mathrm{P}^{\max}_{\text{return}}$ et $\mathrm{P}^{\text{ave}}_{\text{return}}$, afin d'assurer qu'il existe une solution optimale à support fini.

Ces résultats peuvent être étendus au cas d'une flotte de plusieurs navettes.

Proposition 3. *Résoudre le problème $\mathrm{P}^{\text{ave}}_{\text{return}}$ dans ce cas particulier pour S navettes est équivalent à résoudre le problème $\mathrm{P}^{\text{ave}}_{\text{return}}$ dans ce cas particulier pour 1 seule navette.*

Références

[1] Lineu C. Barbosa and Moshe Friedman. Deterministic inventory lot size models – a general root law. *Management Science*, 24(8) :819–826, 1978.

[2] Valentina Cacchiani and Paolo Toth. Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3) :727–737, 2012.

[3] Daniel R. Dooly, Sally A. Goldman, and Stephen D. Scott. TCP dynamic acknowledgment delay (extended abstract) : theory and practice. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 389–398. ACM, 1998.

[4] Leonard Kleinrock. Queueing systems, volume I : theory. 1975.

[5] Paolo Serafini and Walter Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4) :550–581, 1989.