VERS LA ROBUSTESSE EN GESTION DU REVENU : ANALYSE COMPETITIVE DES DECISIONS DE PRIX ET DE RESERVATION

Maurice QUEYRANNE
Sauder School of Business, U. of British Columbia, Vancouver (Canada)
et Laboratoire Leibniz, Institut IMAG, Grenoble (France)

Michael BALL
R. H. Smith School of Business & Institute for Systems Research
University of Maryland

Analyse compétitive

$€ \longleftarrow \boxed{A} \longleftarrow I = (r_1, r_2, r_3, \ldots)$

Concepteur d'algorithme

Algorithme

Séquence de requêtes

« Adversaire »

Rapport compétitif pour l'algorithme A : $\boxed{c_{\mathbf{I}}(A) = \min_{I \in \mathbf{I}} \frac{R(A,I)}{R^*(I)}}$

$\mathbf{I}$ = ensemble de requêtes (ou instances) considérées

$R(A,I)$ = profit lorsque l'algorithme A est appliqué à l'instance I

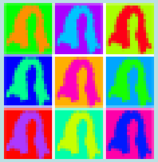
$R^*(I)$ = profit *optimal* pour l'instance I

Exemple de résultats

- Un vol avec :
  - 95 places disponibles
  - 3 classes de prix : 1.000€ , 750€ , 500€
- Politique qui garantit(*) au moins **63%** du revenu maximum possible :
  - Protéger 15 places aux plus haut prix, et
  - Protéger un total de 35 places pour les deux plus haut prix

    (donc : vendre au plus 60 places au plus bas prix)

(*) sous une *Hypothèse des Prix Non-décroissants*



€

Analyse : cas de 2 prix

Intuition :

- Il est toujours optimal d’accepter une requête au plus haut prix (H)

➔L’adversaire pourra commencer avec des requêtes au plus bas prix (B)

Question :

- combien faut-il accepter de B avant de n’accepter que des H ?

Remarque : il faut toujours accepter la première requête (sinon l’adversaire s’arrête et le rapport compétitif est zéro)

Cas de 2 prix : le dilemme

places disponibles

B B B B | B B B B B B B B B B B B

A cesse d'accepter B

A accepte <u>trop peu</u> de B
➔ l'adversaire n'enverra que des B
(jusqu'à capacité)

B B B B B B B B B B B B B | H H H H H H H H H H H H H H

A accepte <u>trop</u> de B
➔ l'adversaire n'enverra que des H
(jusqu'à capacité)

Cas de 2 prix : notations

Notations :

n nombre de places disponibles (capacité à allouer)

$f_H > f_B$ prix

$r = f_B / f_H$ taux de *réduction* $(0 < r < 1)$

$b(r) = 1 / (2 - r)$ ***rapport critique (compétitif)***

solution de l'équation: $y = 1 - (1 - r)y$

Cas de 2 prix : stratégie adverse

Instances spéciales

- $I_1 = n$ B $\qquad R^*(I_1) = f_B\, n = r\; f_H\, n$
- $I_2 = n$ B, suivis de n H $\qquad R^*(I_2) = f_H\, n$

Stratégie adverse contre un algorithme déterministe A :

- envoyer n requêtes B
- si A accepte **moins** de $b(r)\, n$ requêtes alors compléter I_1 :

$$R(A, I_1) < f_H\, b(r)\, n = b(r)\, R^*(I_1)$$

- sinon compléter avec I_2 :

$$R(A, I_2) \le f_H(n - (1 - r)\, b(r)\, n) = b(r)\, R^*(I_2)$$

Proposition:

Pour le problème à 2 classes, aucun algorithme déterministe n'a un rapport compétitif supérieur à $\mathbf{b(r)}$

Cas de 2 prix :
Algorithmes de réservation

Algorithme de réservation $A(y)$

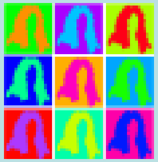
- défini par la fraction y $(0 \le y \le 1)$:

Accepter :

- toutes les requêtes H
  (dans la mesure des places disponibles)
- au plus $y\, n$ requêtes B

De façon équivalente :

- protéger $(1 - y)\, n$ places pour les H



Cas de 2 prix : Algorithme de réservation

Algorithme de réservation $A(y)$

- Supposons $y\,n$ entier :

$y\,n$ $\quad$ n

I_1 : B B B B B B B B B B B B ~~B B B B B B~~

$$R(A(r), I_1) = y\,R^*(I_1)$$

I_2 : B B B B B B B B B B B B ~~B B B B B B~~

H H H H H H H H H H H…

$$R(A(r), I_2) \;=\; y\,f_B\,n + (1-y)\,f_H\,n \;= [1-(1-r)y]\,R^*(I_2)$$

Cas de 2 prix : rapport compétitif d'algorithme de réservation $A(y)$

…et pour les autres inputs I ?

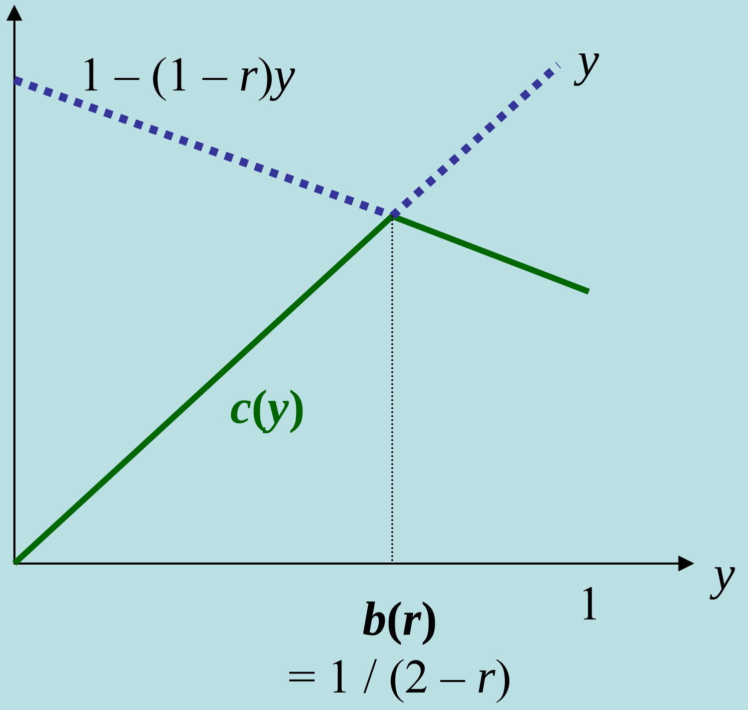
Lemme :

Pour tout Algorithme de Réservation $A(y)$ et tout input I

$$R(A(y), I) \geq c(y)\, R^*(I)$$

où :

$$\mathbf{c(y)} = \min\{\, y\, ,\, 1 - (1-r)y \,\}$$



Cas de 2 prix : rapport compétitif de l'algorithme de réservation $A(b(r))$

Théorème :

Si $b(r)\,n$ *est entier alors l'algorithme* $A(b(r))$ *a un rapport compétitif optimal* $\mathbf{b(r)}$

Exemples : n = 90

r = 75% : places protégées : 18 ; $b(r) = 80\%$

r = 50% : places protégées : 30 ; $b(r) = 67\%$

r = 33% : places protégées : 36 ; $b(r) = 60\%$

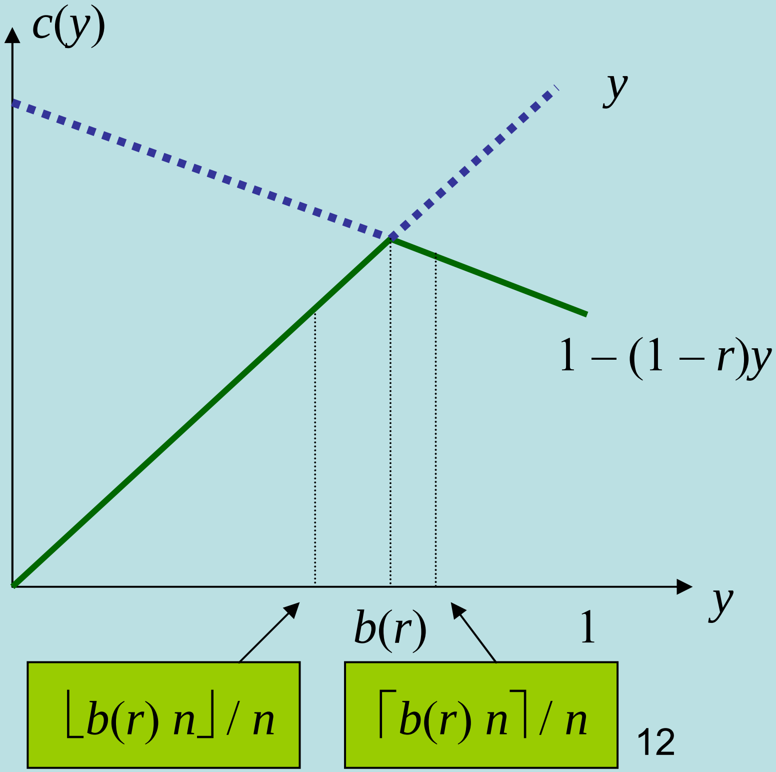
Cas de 2 prix : capacité discrète

Et si $b(r)\, n$ n'est pas entier ?

Soit $\boldsymbol{\beta(r, n)} = \lfloor b(r)\, n \rfloor$ si $c(\lfloor b(r)\, n \rfloor / n) \geq c(\lceil b(r)\, n \rceil / n)$

$\lceil b(r)\, n \rceil$ sinon

Théorème :
Pour le problème discret l'algorithme $A(\beta(r, n))$ a un rapport compétitif optimal parmi tous les algorithmes (déterministes)



Algorithmes aléatoires

Algorithme aléatoire :

- peut utiliser des nombres aléatoires pour ses décisions

Exemple : accepter le i-ème B avec probabilité p_i
chaque H avec probabilité 1

- Le **revenu moyen** de A' pour l'input $I \in \mathbf{I}$ est $\mathrm{E}[R(A', I)]$

- L'algorithme aléatoire A' est **c-compétitif** si

$$\mathrm{E}[R(A', I)] \geq c\, R^*(I) \quad \text{pour tous les inputs } I \in \mathbf{I}$$

Conception d’algorithme aléatoire

Le Concepteur cherche un algorithme qui soit c-compétitif *pour le plus grand rapport c possible* :

$$c^* = \max_{A'} \min_{I \in \mathbf{I}} \frac{\mathrm{E}[R(A', I)]}{R^*(I)}$$

- Cette définition correspond à l’hypothèse d’un **adversaire oublieux** (*oblivious adversary*) qui peut choisir l’input I
  - en connaissant l’algorithme A'
  - mais **pas les réalisations** de ses variables aléatoires

(de façon équivalente :
A' doit générer **tout l’input I avant** d’observer les choix de A')

Algorithmes aléatoires : le principe de Yao

Observation fondamentale (Yao, 1977) :

Un algorithme aléatoire peut être considéré comme un choix aléatoire parmi des algorithmes déterministes

Soients :

$\mathbf{A}$ l'ensemble de tous les algorithmes déterministes

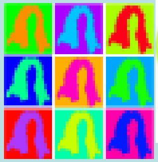
$\mathbf{P}$ l'ensemble de toutes les distributions de probabilité sur $\mathbf{A}$

$A(P)$ l'algorithme aléatoire déterminé par $P \in \mathbf{P}$

$\mathbf{Q}$ l'ensemble de toutes les distributions de probabilité sur $\mathbf{I}$

alors (**principe de von Neuman/Yao**) :

$$c^* = \max_{P \in \mathbf{P}} \min_{I \in \mathbf{I}} \frac{\mathrm{E}_P[R(A(P), I)]}{R^*(I)} \leq \min_{Q \in \mathbf{Q}} \max_{A \in \mathbf{A}} \mathrm{E}_Q\left[\frac{R(A, I(Q))}{R^*(I(Q))}\right]$$



Cas de 2 prix : Algorithmes aléatoires optimaux

Théorème :

Pour le problème à 2 prix, le meilleur rapport compétitif d'un algorithme aléatoire est $\mathbf{b(r)}$

Théorème :

Ce rapport $\mathbf{b(r)}$ *est atteint par tout Algorithme de Réservation aléatoire dont* ***l'espérance*** *du nombre de places protégées est* $1 - b(r)$

Cas général : m classes

m classes de requêtes avec prix :

$$f_1 > f_2 > \ldots > f_m$$

Soit $$\Delta = m - \sum_{i=2..m} f_i / f_{i-1}$$

Théorème :

Pour le problème avec m classes de prix, aucun algorithme, déterministe ou aléatoire ne peut avoir un rapport compétitif plus grand que **1/Δ**

Exemple : f_1 = 1000, f_2 = 750, f_3 = 500, 1 / $\Delta \approx$ 63%

Gestion des prix et des réservations

Les requêtes arrivent avec des prix quelconques
…lesquelles accepter ?

Théorème :

Lorsque les prix sont entre les limites f_{max} *et* $f_{min} = r\, f_{max}$ *aucun algorithme, déterministe ou aléatoire ne peut avoir un rapport compétitif plus grand que* $\mathbf{1 / (1 - \ln r)}$

Exemples: $r = 3/4 : \ 1 / (1 - \ln r) \approx 0{,}78$

$r = 1/2 : \ 1 / (1 - \ln r) \approx 0{,}59$

$r = 1/3 : \ 1 / (1 - \ln r) \approx 0{,}48$

Gestion des prix et des réservations : les politiques $P(\rho)$

Soit ρ un paramètre satisfaisant

$$0 < \rho < 1/(1 - \ln r)$$

Politique $P(\rho)$:

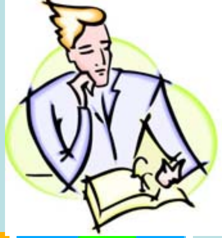
- accepter les $\lceil \rho\, n \rceil$ premières requêtes (tant que leur prix $f \geq f_{min}$)
- ensuite accepter chaque requête si son prix

$$f \geq f_{min}\,(1 + 1/\,(\rho\, n))^{\lceil a - \rho\, n - 1 \rceil}$$

où $a \geq \rho\, n + 1$ est le nombre de requêtes déjà acceptées

Remarque : le taux de refus augmente avec a et avec ρ

Gestion des prix et des réservations : rapport compétitif des politiques $P(\rho)$



Théorème :

Pour tout niveau cible de performance ρ, satisfaisant

$$0 < \rho < 1 / (1 - \ln r)$$

si la capacité satisfait

$$n \geq \mathbf{g(r, \rho)}$$

alors le rapport compétitif de la politique $P(\rho)$ est au moins

$$\rho\,\mathbf{(1 - 1 / (\rho\,(\rho n + 1)))}$$

Remarques :

- la fonction $g(r, \rho)$ est assez compliquée
- la performance de $P(\rho)$ peut être proche de la borne supérieure $1 / (1 - \ln r)$ quand n est suffisamment grand

Gestion des prix et des réservations : propriétés des politiques $P(\rho)$

Exemple :

Soit r = 0,5 alors $1/(1 - ln\, r) \approx$ 0,59

- pour ρ = 0,58 on doit avoir $n \geq$ 19,002
  - avec n = 20, (1 – 1 / (ρ (ρ *n* + 1))) = 0,863
  - avec n = 100, (1 – 1 / (ρ (ρ *n* + 1))) = 0,970

…mais la borne inférieure $g(r, \rho)$ sur n augmente quand ρ s'approche de $1/(1 - ln\, r)$

En modifiant ρ on peut ajuster la garantie de performance et le taux de refus de la politique $P(\rho)$

Conclusion et perspectives

- L'approche compétitive peut s'appliquer en *l'absence d'information sur la demande*
- Les politiques résultantes sont raisonnables
- L'approche compétitive peut aussi suggérer des *nouvelles classes de politiques*, par exemple :
  - Les méthodes classiques de Gestion du Revenu pour le transport aérien supposent souvent que les requêtes arrivent avec des prix non-décroissants
  - Les politiques de type $P(\rho)$ permettent des ajustements plus dynamiques :