Revenue Management sur un marché

*Synthèse – avancement du projet au 25 Mars 2020*

Barras François – Akani Julian – Gaillard Charles

*Modélisation du simulateur de vols + clients*

Langage : Python

Il y a 2 classes dans notre modèle : **Vol** et **Client**

Un **vol** a pour attributs :

* Heure dans la journée
* Nombre de places
* Vecteur des places déjà vendues sur ce vol (contenant le prix de chaque place vendue),
* Prix de la place la moins chère sur ce vol qui sera mise à jour après chaque client et proposé au client suivant
* Utilité horaire qu’ont les clients pour ce vol en raison de son positionnement dans la journée (on considère que tous les clients ont les mêmes préférences horaires pour les vols).

Un **client** a pour attributs :

* Jour de son arrivée sur le site (par rapport au départ du vol)
* Sensibilité au prix (qui elle, contrairement à la sensibilité horaire va dépendre du jour d’arrivée du client)

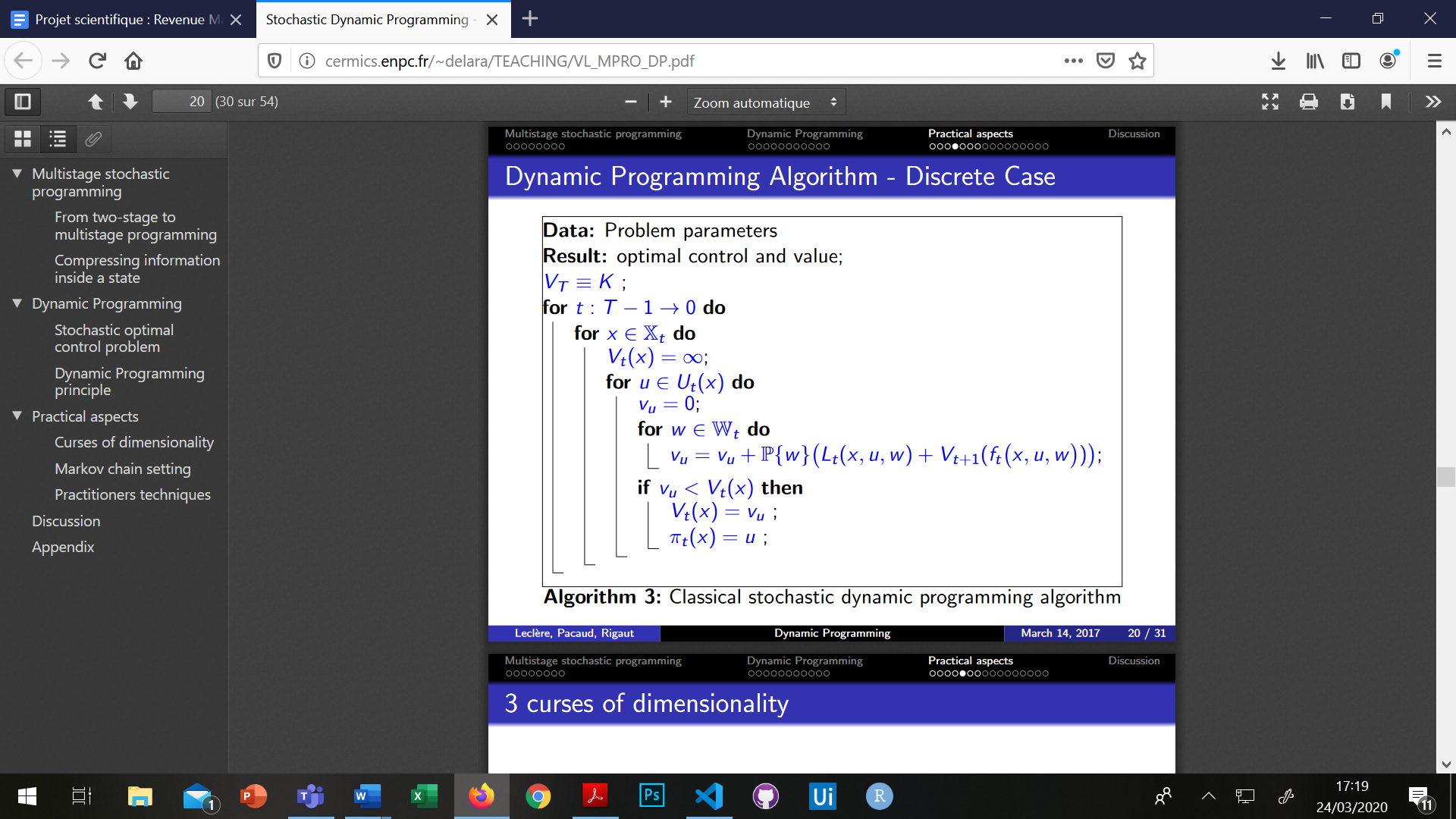
On va générer un ensemble de vols sur une journée, et un ensemble de clients. On considérera ensuite les clients 1 par 1 dans l’ordre chronologique des jours.

Nous adoptons donc une approche par clients : chaque pas de temps correspond à l’arrivée d’un client sur le site de la compagnie aérienne. Ce client se voit proposer un prix pour chaque vol de la journée. Il va alors, en fonction de son utilité pour chaque vol (prix, heure du vol) choisir : soit 1 de ces vols, soit de ne pas acheter de billet.

Une fois que le client achète ou non son billet, on met à jour le vecteur des places vendues pour le vol en question, et surtout on ajuste le prix des places restantes que l’on va proposer aux futurs clients.

Pour trouver le pricing optimal, nous utilisons un algorithme de **Programmation dynamique stochastique.**

* Etape **t = 1, … , n** = ensemble des pas de temps.
* Etat **xt** = Pour chaque vol, sièges déjà vendus.
* Décision **ut** = S’il reste des sièges dans les vols, on fixe des prix pour chaque vol. S’il ne reste plus aucun siège, on vient de définir une politique de prix possible.
* Choix du client **wt**= le client choisi un des vols ou aucun vol (suite au pricing ut).
* Transition de l’état xt+1 à l’état xt grâce à la fonction **ft(xt,ut,wt)**
* **vu** (Espérance de gain associée à la décision ut): espérance des revenus générés par la décision de pricing ut
* **Lt** fonction bénéfice (que l’on cherche à maximiser)



**Modèle d’utilité et choix du client :**

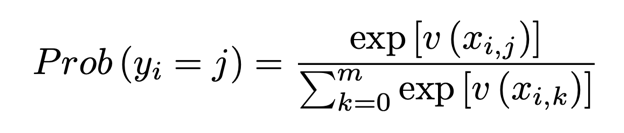
On propose au client différents vols et celui-ci va acheter l’un d’entre eux ou pas en maximisant son utilité, qui est une fonction positive ou nulle des caractéristiques d’un vol. On modélise l’utilité d’un client i pour un vol j de caractéristiques xj selon une partie déterministe et un bruit stochastique :

Ui(xj) = Vi(xj)+ εj

La fonction Vi s’exprime comme la somme de deux fonctions à valeurs positives ou nulles : l’utilité horaire et de la sensibilité au prix. Cette somme est pondérée par un facteur d’échelle θ entre les 2:

Vi(xj)= Hi(xj) + θ\*Si(xj)

On représente le choix du client i par la variable yi. yi =0 si le client n’achète aucun vol. yi =j si le client choisit le vol j. On adopte alors le modèle logit multinomial non ordonné selon lequel la loi de yi est :



Cette loi nous permet de simuler le choix du client. Il nous reste à fixer un paramètre : l’utilité associée à un refus d’acheter, qui nous est inconnue. Celle-ci est équivalente à l’utilité d’acheter chez la concurrence ou de ne pas voyager. Elle est non nulle.

**Utilité horaire des clients (sur 15h : de 7h à 22h)**

Pic le matin et pic plus applati l’après-midi/soir avec un creux pour le diner (typiquement sur un vol court de type Paris-toulouse, Paris-Londres).



**Sensibilité au prix**

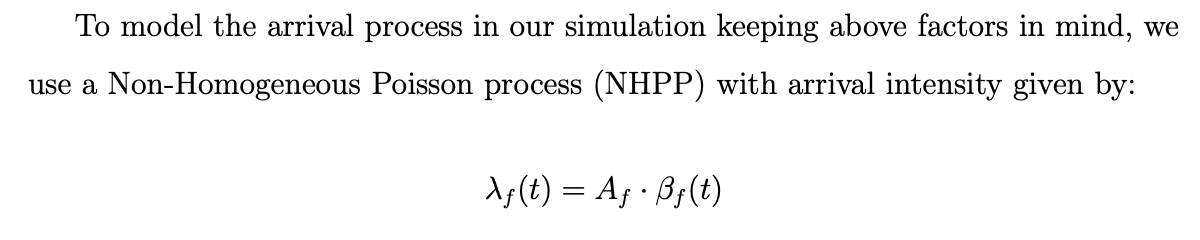
Fonction à valeurs dans [0,1]. On peut par exemple utiliser une fonction exponentielle décroissante du prix du vol pj, qui tend vers une utilité nulle quand les prix tendent vers l’infini, de coefficient variable du temps α(t) :

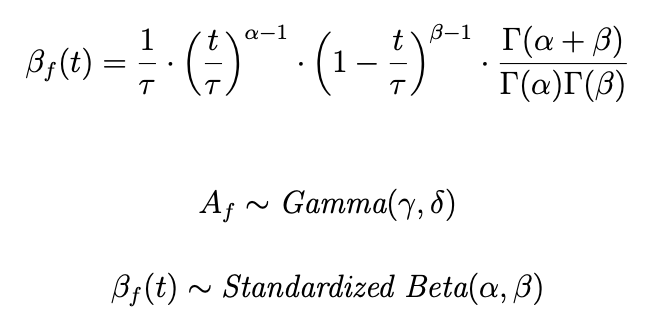
Si(xj) = exp(-α(t)\*pj)

Et l’on peut supposer par exemple que le coefficient α(t) croît linéairement lorsqu’on s’approche du jour de départ entre 2 valeurs αmin (ouverture de la billetterie) et αmax (jour du départ).

**Répartition temporelle de l’arrivée des clients**

On utilise un processus de Poisson non homogène pour modéliser le nombre de clients qui arrivent chaque jour. L’intensité de la loi de Poisson s’écrit λ(t) et dépend du temps selon la loi suivante[[1]](#footnote-1) :

****

****

Les paramètres de la loi beta sont choisis selon le type de client considéré (business, economy…)

1. https://pdfs.semanticscholar.org/2a45/750ff0dffe131a02f92a9a66f8b678b0e057.pdf [↑](#footnote-ref-1)