

CL04/4E: Projet P2024

Gestion d'un Stock dans un Magasin Omnicanal

Un magasin omnicanal vend un produit à la fois hors ligne (dans le magasin) et en ligne, donc il y a deux classes (types) de demande pour ce produit. La demande hors ligne du produit suit un processus de Poisson de paramètre λ_1 , alors que la demande en ligne suit un processus de Poisson de paramètre λ_2 , où λ_1 (resp. λ_2) est le nombre moyen d'arrivée de demande unitaire hors ligne (resp. en ligne) dans chaque jour. Ce magasin est ouvert tous les jours. Il s'approvisionne en ce produit auprès d'un fournisseur avec le délai d'approvisionnement de L jours. On suppose que ce fournisseur n'est jamais en rupture de stock de ce produit, c'est-à-dire, le magasin reçoit la livraison de sa commande passée au fournisseur juste après L jours de sa passation de cette commande. Le magasin fait l'inventaire du produit continuellement (en temps réel). Il emploie une politique, notée politique (Q, r, K) , pour gérer l'approvisionnement de son stock du produit et pour rationner ce stock entre la demande hors ligne et la demande en ligne, $Q > 0, r \geq K \geq 0$. A tout moment lorsque le stock disponible du stock descend à r , le point de commande, une commande de Q unités sera passée au fournisseur pour l'approvisionnement du stock. Le paramètre K est le niveau (seuil) critique pour le rationnement entre les deux classes de demande.

Lorsque le stock physique du produit de ce magasin est supérieur à K , si une unité de demande hors ligne ou une commande en ligne d'une unité du produit arrive, cette demande ou commande sera satisfaite ou livrée au client immédiatement suivant la règle FIFO (First In First Out) ou la règle « premier arrivée premier satisfaite ». Lorsque ce stock physique est inférieur ou égal à K , si une unité de demande hors ligne arrive quand un client visite le magasin, cette demande sera satisfaite immédiatement si ce produit est disponible dans le magasin (c'est-à-dire, le stock physique ≥ 1) ; sinon cette demande sera perdue à cause de la rupture de ce produit dans le magasin, engendrant un coût de perte de vente. En revanche, lorsque le stock physique est inférieur ou égal à K , si une unité de demande en ligne arrive quand un client passe une commande d'une unité du produit en ligne, cette commande (une unité du produit) sera livrée du magasin au client immédiatement ; sinon la livraison de cette commande est différée (retardée) jusqu'au moment où ce stock physique s'élève à un niveau supérieur à K suite à l'arrivée à ce magasin d'un nouvel approvisionnement du produit. Le magasin promet à ses clients en ligne un délai de livraison de W jours pour chaque commande. Si la livraison d'une commande dépasse ce délai, une indemnité sera payée au client concerné, cette indemnité est calculée par le nombre de jours retard multiplié par l'indemnité journalière de retard. On suppose que le temps de livraison de chaque commande est négligeable.

On suppose que lorsque le magasin passe une nouvelle commande d'approvisionnement au fournisseur, toutes les commandes passées au fournisseur auparavant sont déjà livrées au magasin, c'est-à-dire, le magasin a une commande à exécuter au maximum.

L'objectif du magasin est de minimiser le coût total journalière en moyenne du stock de ce produit comprenant ses coûts de passation de commande, coûts de stockage, coûts de perte de vente hors ligne, et coûts d'indemnité de livraisons retardées de commandes en ligne. Les paramètres de coût sont définis comme suit :

F : coût de passation de chaque commande d'approvisionnement par le magasin,

h : coût de stockage par unité du produit par jour dans le magasin,

p : coût de perte de chaque demande unitaire pour le magasin,

b : l'indemnité journalière de retard payée par le magasin pour la livraison retardée d'une commande en ligne.

L'instance à étudier dans ce projet a les paramètres suivants :

$\lambda_1 = 15, \lambda_2 = 30, L = 1, W = 1, F = 18, h = 0.05, p = 20, b = 5, Q = 180, r = 60, K = 10$.

Ce projet a les tâches suivantes :

- 1) Programmez en langage de votre choix (Matlab, C/C++, Python, etc) un outil (une programme) de simulation pour simuler l'évolution (la dynamique) du stock de ce produit dans le magasin pendant un grand nombre de jours durant lesquels au moins 10000 commandes d'approvisionnement du stock sont passées par le magasin pour obtenir des résultats de simulation fiables. Au début de la simulation de ce stock, on suppose que son stock physique est égal à r et il n'y pas de commander à exécuter ni demande insatisfaite.
- 2) Pour l'instance donnée et basé sur les résultats de la simulation effectuée dans 1), estimez le niveau moyen de stock, le pourcentage de perte de demande hors ligne, le pourcentage de commandes en ligne livrées en retard, et le coût total journalière en moyenne du stock.
- 3) Variez la valeur de K de 0 à r sans changer la valeur de Q et de r et analysez l'impact de K sur les indicateurs de performance étudiés dans 2). Pour cette analyse, il faut simuler le stock pour chaque valeur de K considérée.
- 4) Rédigez un petit rapport de 5 pages au maximum avec votre code informatique mis dans une annexe. Dans ce rapport, vous présentez vos résultats obtenus pour 2) et votre analyse de l'impact de K pour 3).

Le projet peut être fait tout seul ou en binôme. Si vous choisissez de faire ce projet en binôme, vous pouvez choisir votre partenaire librement, mais chaque étudiant(e) du binôme doit participer à la fois à la programmation informatique et à la rédaction du rapport. De plus, la contribution de chaque étudiant(e) à chaque tâche doit être bien précisée dans le rapport. La réalisation de ce projet tout seul est encouragée avec un bonus – la majoration de la note du projet par 10%.

Il faut me rendre par courriel votre rapport du projet en un seul fichier PDF nommé par "projetCL04_votre nom" ou "projetCL04_vos deux noms" avant l'examen final de CL04. Veuillez écrire le sujet de votre courriel comme le nom du fichier.

Indications :

- 1) Si la demande hors ligne (resp. en ligne) du produit suivre un processus de Poisson de paramètre λ_1 (resp. λ_2), le temps entre deux arrivées consécutives de demande unitaire hors ligne (resp. en ligne) est soumise à une loi exponentielle de paramètre λ_1 (resp. λ_2).
- 2) L'arrivée de demandes unitaires de deux classes (hors ligne et on ligne) du produit dans le magasin peut être décrite de manière équivalente par deux lois : une loi exponentielle de paramètre $\lambda_1 + \lambda_2$ et une loi de Bernoulli avec la probabilité $\lambda_1/(\lambda_1 + \lambda_2)$ pour la demande hors ligne et la probabilité $\lambda_2/(\lambda_1 + \lambda_2)$ pour la demande en ligne. C'est-à-dire, le temps entre deux arrivées consécutives de demande unitaire quelconque est soumise à une loi exponentielle de paramètre $\lambda_1 + \lambda_2$. mais pour chaque demande unitaire qui arrive, elle serait une demande hors ligne avec la probabilité $\lambda_1/(\lambda_1 + \lambda_2)$ et une demande en ligne avec la probabilité $\lambda_2/(\lambda_1 + \lambda_2)$.

Références :

https://fr.wikipedia.org/wiki/Processus_de_Poisson
https://fr.wikipedia.org/wiki/Loi_de_Poisson
https://fr.wikipedia.org/wiki/Loi_exponentielle
https://fr.wikipedia.org/wiki/Loi_de_Bernoulli