Projet GP30 : Reproduction et analyse d'un article scientifique

Nathan Davouse, Alex Perron - Semestre de Printemps 2024

1 Résumé du problème

1.1 Description de la problématique

Les produits frais se détériorent au fil du temps, ce qui entraîne une diminution de leur qualité et de leur valeur. Ainsi, il est important de réfléchir à un autre moyen de les vendre.

La stratégie de regroupement présente plusieurs avantages et est un moyen efficace de promouvoir la vente. Premièrement, elle permet de proposer les produits à un prix plus bas et donc plus attrayant pour plus de consommateurs. Ainsi, un lot de 2 produits sera vendu moins cher que 2 produits vendus séparément. Deuxièmement, elle incite les clients à consommer plus et permet d'augmenter la vitesse de vente. Enfin, elle permet une gestion différente de la qualité et fraîcheur des produits, notamment avec la possibilité de proposer des réductions sur les packs moins frais afin d'éviter le gaspillage.

Cependant, cette stratégie d'emballage n'est pas simple à appliquer. En effet, il faut décider de la valeur de plusieurs paramètres pour créer un lot : il doit regrouper la bonne quantité de produits, avoir un prix approprié à sa fraîcheur et être proposé au bon moment. Tous ces paramètres doivent être choisis pour maximiser le profit du vendeur mais ils visent également à satisfaire les exigences de qualité des consommateurs.

L'article étudié [1] propose donc un modèle mathématique permettant de prendre des décisions sur les paramètres exprimés ci-dessus pour des produits frais avec une détérioration de leur qualité au cours du temps. Les hypothèses sont les suivantes :

— les lots sont composés de produits de même fraîcheur ;
— les lots sont constitués d'un unique type de produit ;
— la valeur des produits frais en fonction de leur détérioration est représentée par la fonction exponentielle suivante : $V(t) = e^{-b(t-1)}$ où b représente le facteur de diminution du produit frais ($b > 0$) ;
— le problème est envisagé du point de vue du vendeur.

1.2 Formulation et explication des contraintes

Pour répondre à la problématique, nous pouvons utiliser le modèle défini par l'article présenté. Ce modèle reprend l'ensemble des éléments définis plus haut, et les variables de décisions sont le regroupement choisi par le commerçant Y_{jt}, le prix de ce regroupement P_{jt}, la décision d'acheter le regroupement par le client X_{ijt} et le surplus par client, soit le gain qu'il effectue lorsqu'il achète le produit S_i.

Pour se concentrer sur le modèle, nous allons détailler le fonctionnement et l'utilité de chaque contrainte :

— (3) $\max \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{t=1}^{T} (P_{jt} - c_j) X_{ijt}$ Cette fonction est la fonction objective, qui cherche à maximiser le profit du commerçant. Le profit est exprimé comme le prix de vente moins le coût de production pour chaque achat qu'un client effectue.
— (4) $S_i = \sum_{j=1}^{J} \sum_{t=1}^{T} (R_{ijt} - P_{jt}) X_{ijt}, \forall i$. Cette contrainte définie le surplus du consommateur, soit le gain qu'il fait en achetant le pack au prix proposé par rapport au prix maximum qu'il s'attendait à payer.
— (5) $S_i \geq (R_{ijt} - P_{jt}) Y_{jt}, \forall i; \forall j; \forall t$. Cette contrainte permet à chaque client de maximiser son surplus. Cela vérifie que le surplus correspond au choix est bien le surplus maximum possible pour le client.

1. Bundle Pricing Decisions for Fresh Products with Quality Deterioration; Yan Fang et al

- (6) $(R_{ijt} - P_{jt})X_{ijt} \geq 0, \forall i; \forall j; \forall t$. Cette contrainte représente la rationalité du client : il ne choisira un lot que si son surplus est positif.
- (7) $R_{ijt} = \left[jR_{i11} - \beta(j-1)^2\right] \cdot e^{-b(t-1)}$ Cette contrainte permet de calculer le contenu de la matrice R_{ijt}, c'est à dire le prix de réservation par le client i du lot de taille j au moment t. Ce "prix de réservation" correspond à la limite supérieure que le client est prêt à payer pour un produit (willingness to pay). Dans le cas envisagé ici, des lots de plusieurs produits frais, les coûts de réservation diminuent en accord avec la perte de fraîcheur des produits. Ces coûts prennent également en compte la loi de l'utilité marginale décroissante, c'est à dire que le ratio R_{ijt}/j diminue avec l'augmentation de la taille du lot.
- (8) $\sum_{j=1}^{J} \sum_{t=1}^{T} X_{ijt} \leq 1, \forall i$. Cette contrainte empêche les clients de réserver plus d'un lot chacun.
- (9) $\sum_{j=1}^{J} Y_{jt} \leq 1, \forall t$. Cette contrainte empêche le vendeur de proposer plus d'un lot différent par période.
- (10) $X_{ijt} \leq Y_{jt}, \forall i; \forall j; \forall t$. Cette contrainte permet de ne rendre l'achat possible par les clients d'un certain lot que s'il est rendu disponible par le vendeur.
- (11) et (12) : Ces contraintes empêchent les variables S_i et P_{jt} de prendre des valeurs négatives. Cela semble logique pour le prix de vente et le surplus client.
- (13) et (14) : Ces contraintes assurent que les variables X_{ijt} et Y_{jt} soient binaires.

1.3 Méthodes de résolution utilisées

Le problème de ce modèle mathématique est qu'il est non linéaire et donc assez compliqué à résoudre. L'approche pour simplifier la résolution est de le convertir en modèle linéaire. Pour cela, il faut effectuer un changement au niveau des variables de décision. Les variables $K_{ijt} = P_{jt}X_{ijt}$ et $L_{jt} = P_{jt}Y_{jt}$ sont donc introduites afin d'éviter les multiplications entre variables de décision, ce qui rendait le problème non linéaire.
Ces nouvelles variables impactent directement le modèle. En effet, il est important de garantir que K_{ijt} et L_{jt} prennent bien les valeurs voulues. Pour cela, un big M est introduit. Il est défini dans les contraintes du modèle : (15) $M = 10 \sum_{j=1}^{J} c_j$. De nouvelles contraintes utilisent ce big M pour encadrer la valeur des deux nouvelles variables de décision et forcer leur valeur. Par exemple, pour encadrer K_{ijt} les contraintes suivantes sont utilisées :

- (3.2) $K_{ijt} - M(1 - X_{ijt}) \leq p_{jt}, \forall i.; \forall j.; \forall t$. Dans le cas ou $X_{ijt} = 1$, elle force $K_{ijt} \leq p_{jt}$.
- (3.3) $K_{ijt} + M(1 - X_{ijt}) \geq p_{jt}, \forall i.; \forall j.; \forall t$. Dans le cas ou $X_{ijt} = 1$, elle force $K_{ijt} \geq p_{jt}$.
- (3.4) $K_{ijt} - MX_{ijt} \leq 0, \forall i.; \forall j.; \forall t$.Dans le cas ou $X_{ijt} = 0$, elle force $K_{ijt} \leq 0$.
- (16) $K_{ijk} \geq 0$ Cette contrainte force $K_{ijt} \geq 0$.

Ces contraintes traduisent une disjonction des cas :

- dans le cas $X_{ijt} = 1$, alors $P_{jt}X_{ijt} = P_{jt}$. Dans ce cas, les contraintes (3.2) et (3.3) permettent d'obtenir $K_{ijt} = P_{jt} = P_{jt}X_{ijt}$.
- dans le cas $X_{ijt} = 0$, alors $P_{jt}X_{ijt} = 0$. Dans ce cas, les contraintes (3.4) et (16) affectent K_{ijt} tel que $K_{ijt} = 0 = P_{jt}X_{ijt}$.

De cette façon, la relation $K_{ijt} = P_{jt}X_{ijt}$ est bien garantie.
Le même principe de linéarisation est utilisé pour garantir la valeur de la variable L_{jt}. Ainsi, ce sont les contraintes (5.2), (5.3), (5.4) et (17) qui contraignent L_{jt}, de sorte à ce que $L_{jt} = P_{jt}.Y_{jt}$.

Une fois la linéarisation effective, l'enjeu est la programmation de ce modèle pour assurer une résolution. Dans un premier temps, le modèle non-linéaire (modèle non_linear_problem.lg4) ayant été réalisé sur Lingo, nous avons réalisé le modèle linéaire sur Lingo (modèle linear_problem.lg4). Cependant, Lingo n'était pas adapté à un modèle linéaire de cette complexité ; les temps de calcul était trop long et dépendant de la génération aléatoire pour pouvoir correctement explorer et exploiter les résultats. C'est pour cela que nous avons travaillé sur un modèle utilisant Gusek et son solveur GLPK (modèle linear_model.mod). Malheureusement, même si Gusek est fait pour les modèles linéaires, il n'était pas suffisamment puissant. C'est pour cela que nous avons réalisé le modèle sur Gurobi, utilisant l'API sur python et sa librairie, gurobipy (modèle bundle_model.py). En plus d'assurer un temps de calcul raisonnable (moins d'une minute par résolution), l'appel par python (fichier sensi_analysis.py) permet d'automatiser les tests effectués par la suite.

2 Analyse des résultats

2.1 Étude avec les données d'origine

Dans cette partie, nous utilisons le modèle linéaire du 1.3 pour étudier les résultats et l'effet des différents paramètres. Sur la table suivante, nous avons la valeur des paramètres initiaux de l'instance.

I	J	T	c_j($)	R_{i11}($)	β	b
10	10	6	$4j$	$U(6,12)$	0,5	0,04

Table 1 – Valeurs initiales des paramètres

Dans la table 1, les 3 paramètres clés que sont β, b et R_{i11} influent sur les valeurs de R_{ijt}, le prix de réservation. Pour rappel, β affecte la réduction liée au groupement de produit, b affecte la réduction liée au temps qui passe, et R_{i11} définit le prix de réservation initial du client i. C'est intéressant, puisque que les paramètres que l'on cherche à étudier ne représentent que le comportement de l'utilisateur, le reste est fixé ou s'adapte sur l'utilisateur.

2.1.1 Étude du résultat d'une instance

Pour comprendre les résultats et le fonctionnement du modèle, nous étudions tout d'abord les résultats pour une instance donné, avec les paramètres de la table 1. Ces résultats sont cependant fortement dépendant de l'aléatoire généré par la fonction uniforme concernant le prix de réservation. C'est pour cela que la valeur de génération aléatoire (*random seed*) est fixé sur cette table, pour permettre de revenir au même résultats.

t	Taille du pack	Prix du pack	Prix unitaire	Nombres d'acheteurs	Profit	Surplus
1	8	59,94 $	7,49 $	4,0	126,40 $	25,86 $
2	3	26,65 $	8,88 $	1,0		
3	3	30,59 $	10,20 $	//		
4	5	49,64 $	9,93 $	//		
5	0	0,0 $	0,0 $	//		
6	1	9,59 $	9,59 $	//		

Table 2 – Résultats de la simulation initiale (*random seed* 78794)

Par rapport à l'article original, nous avons fait le choix d'afficher le nombre de clients qui ont achetés le pack. Cela permet de comprendre les choix de packs proposés. De la sorte, nous pouvons voir que malgré des packs proposés à 5 périodes sur 6, seuls 5 clients achètent et seulement sur les 2 premières périodes. De cette façon, les packs proposés aux périodes 3 au 6 ne sont que peu important, comme ils sont moins avantageux pour les clients que les précédents.

2.1.2 Etude de la distribution des résultats en fonction de la génération aléatoire de R_{i11}

Le prix de réservation R_{ijt} est basé sur une valeur unique par client i, R_{i11}. Ces valeurs sont générés par une loi uniforme de paramètres (6 ;12). Cette loi est dotée d'une moyenne de 9, et d'un écart type d'environ $1,7$.

La figure 1 représente la distribution du profit après 100 tirages aléatoires du vecteur R_{i11}. Ce profit a pour moyenne $114,30$ $ et comme médiane $113,95$ $. Pour les paramètres équivalents, l'article annonce un profit de $156,58$$, ce qui parait être relativement loin de la médiane. On peut considérer leur résultat comme l'un des meilleurs qu'ils ont trouvés lors de leur simulation.

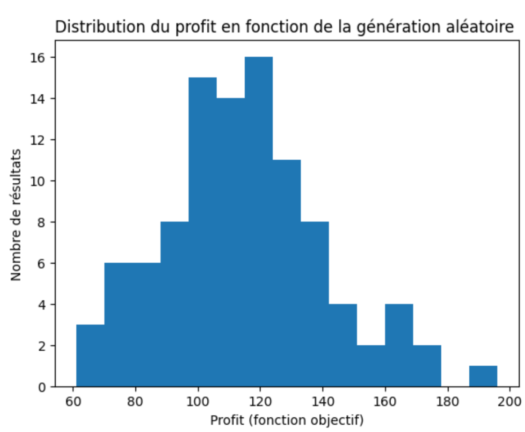

Figure 1 - Distribution du profit

2.2 Analyses de sensibilité

Comme précisé auparavant, les 3 paramètres clés sont β, b et R_{i11}. Par la suite, nous allons étudier l'influence de ces 3 paramètres sur nos résultats, afin de pouvoir conclure sur la pertinence du choix des paramètres. Pour conserver un point de comparaison avec les premiers résultats, la valeur de génération aléatoire utilisée lors de la partie 2.1.1 est conservé pour la suite.

Pour permettre aux tableaux de données par la suite d'être compact, les intitulés des 4 colonnes de la table 2 seront abrégés pour donner BS (Bundle size), BP (Bundle price), BP/PS (Bundle size / Bundle price : prix unitaire) et CN (Consumers numbers).

2.2.1 Analyse de β

Lors de l'optimisation initiale, β avait pris la valeur de 0.5. Pour compléter cette valeur de la même façon que l'article, nous allons comparer la valeur initiale avec des simulations pour $\beta = [0,2; 0,8]$. Comme vu dans la contrainte (7), β pondère la perte de valeur des produits aux yeux des clients lorsqu'ils sont regroupés. De cette façon, plus β est conséquent, moins les clients achèteront des packs à un prix élevé.

β	$\beta = 0,2$				$\beta = 0,5$				$\beta = 0,8$			
t	BS	BP	BP/BS	CN	BS	BP	BP/BS	CN	BS	BP	BP/BS	CN
1	10	83,06 $	8,31 $	5,0	8	59,94 $	7,49 $	4,0	5	34,64 $	6,93 $	3,0
2	2	17,44 $	8,72 $	1,0	3	26,65 $	8,88 $	1,0	2	13,45 $	6,73 $	6,0
3	5	41,90 $	8,38 $	//	3	30,59 $	10,20 $	//	10	164,26 $	16,43 $	//
4	0	0,00 $	0,00 $	//	5	49,64 $	9,93 $	//	10	157,82 $	15,78 $	//
5	0	0,00 $	0,00 $	//	0	0,00 $	0,00 $	//	10	151,63 $	15,16 $	//
6	0	0,00 $	0,00 $	//	1	9,59 $	9,59 $	//	10	145,69 $	14,57 $	//
Profits	224,73 $				126,40 $				76,64 $			

Table 3 – Analyse de sensibilité de β (*random seed* 78794)

Sur la table 3, nous pouvons déjà observer le fait que le profit est inversement proportionnel à β. En rentrant dans le détail, le premier enjeu fort est le regroupement et le prix proposés à la première période. Un β avec une valeur faible permet de faire un rassemblement important, à un prix unitaire élevé.

La stratégie du commerçant vis à vis de β est de proposer des gros packs lorsque β est faible, puisqu'il peut proposer un prix important avec ce pack.

2.2.2 Analyse de b

Le paramètre b est le paramètre qui concerne la dégradation de la valeur du produit aux yeux du clients au cours du temps, défini par la contrainte (7). Un paramètre b avec une forte valeur implique une forte dégradation au cours du temps. Dans cette situation, le commerçant propose un regroupement plus fort dans la première période, pour subir le moins possible l'effet de b. Sur ce tableau, nous pouvons voir quelques prix fixés à 2200$: cela correspond au grand M établi par la contrainte (15), soit le prix maximum possible.

Sur nos résultats présents dans la table 4, la tendance qui est légèrement visible est de pouvoir vendre plus de produit au cours des périodes plus tard lorsque b est faible. Elle se voit principalement sur le nombre d'acheteur lors de la première période par rapport à la suite.

b	$b = 0,01$				$b = 0,04$				$b = 0,07$			
t	BS	BP	BP/BS	CN	BS	BP	BP/BS	CN	BS	BP	BP/BS	CN
1	5	36,92 $	7,38 $	6,0	8	59,94 $	7,49 $	4,0	7	49,56 $	7,08 $	3,0
2	8	53,93 $	6,74 $	2,0	3	26,65 $	8,88 $	1,0	4	29,29 $	7,32 $	2,0
3	2	16,52 $	8,26 $	1,0	3	30,59 $	10,20 $	//	1	7,30 $	7,30 $	1,0
4	6	2200 $	366,67 $	//	5	49,64 $	9,93 $	//	6	2200 $	366,67 $	//
5	1	2200 $	2200 $	//	0	0,00 $	0,00 $	//	6	32,42 $	5,40 $	//
6	5	2200 $	440,00 $	//	1	9,59 $	9,59 $	//	0	0,00 $	0,00 $	//
Profits	153,93 $				126,40 $				94,56 $			

Table 4 – Analyse de sensibilité de b (*random seed* 78794)

2.2.3 Analyse de R_{i11}

Sur l'article qui sert de base à ce rapport, l'action de R_{i11} est étudié en faisant évoluer la borne basse de la loi uniforme permettant de générer . C'est à dire que la génération devait se faire avec la loi uniforme de paramètres {(2,4 ;12) ;(6 ;12) ;(9,6 ;12)}. De cette manière, il y a 2 éléments qui changent avec les données : l'espérance et la variance. Pour dissocier les deux, et permettre une conclusion plus précise, nous proposons d'étudier une évolution de la variance seulement, pour en tirer des conclusions sur l'homogénéité ou non des clients, comme l'article le fait. Comme nous avons vu sur la partie 2.1.2 que les paramètres de bases induisent déjà une forte part d'aléatoire, nous allons tester en réduisant la variance, soit les paramètres {(6 ;12) ;(7 ;11) ;(8 ;10)}.

R_{i11}	$U(6,12)$				$U(7,11)$				$U(8,10)$			
t	BS	BP	BP/BS	CN	BS	BP	BP/BS	CN	BS	BP	BP/BS	CN
1	8	59,94 $	7,49 $	4,0	7	46,73 $	6,68 $	4,0	6	38,58 $	6,43 $	5,0
2	3	26,65 $	8,88 $	1,0	2	13,24 $	6,62 $	5,0	4	26,69 $	6,67 $	5,0
3	3	30,59 $	10,20 $	//	4	26,58 $	6,64 $	1,0	6	35,25 $	5,87 $	//
4	5	49,64 $	9,93 $	//	7	40,63 $	5,80 $	//	0	0,00 $	0,00 $	//
5	0	0,00 $	0,00 $	//	10	57,60 $	5,76 $	//	0	0,00 $	0,00 $	//
6	1	9,59 $	9,59 $	//	2	17,29 $	8,64 $	//	0	0,00 $	0,00 $	//
Profits	126,40 $				111,69 $				126,35 $			

Table 5 – Analyse de sensibilité de R_{i11} (*random seed* 78794)

Les résultats de la table 5 nous montre tout d'abord qu'il n'est pas possible de généraliser sur un lien entre le profit et l'intervalle de la loi uniforme. Cependant, lorsqu'on réduit l'intervalle, les clients ont logiquement tendance à se regrouper, ce qui se voit sur la colonne CN. Sur les paramètres d'origine, seul 5 clients font des achats, car les 5 restants ont des prix de réservation trop bas pour effectuer un achat. Lorsque l'on réduit un peu l'intervalle, on récupère l'ensemble des clients, mais dispersés sur 3 offres. Et en réduisant encore plus, tous les clients sont regroupés sur 2 offres, avec un profit qui revient à l'équivalent du premier. Les prix unitaires sont cependant bien plus bas que la première instance.

Si on résume tout cela, nous pouvons voir qu'au delà du prix, qui dépend de l'aléatoire, nous avons une tendance qui se dessine en fonction de l'intervalle : lorsque les clients sont fortement hétérogènes, le commerçant cherche à faire son profit sur peu de clients, mais des clients prêt a payer assez cher, alors que sinon, l'enjeu est de proposer des prix que tous peuvent s'offrir.