Titre: Title:	A fractional programming approach for choice-based network revenue management
Auteur: Author:	Morad Hosseinalifam
Date:	2009
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Hosseinalifam, M. (2009). A fractional programming approach for choice-based network revenue management [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/8435/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/8435/
Directeurs de recherche: Advisors:	Gilles Savard
Programme: Program:	Génie industriel

UNIVERSITÉ DE MONTRÉAL

A FRACTIONAL PROGRAMMING APPROACH FOR CHOICE-BASED NETWORK REVENUE MANAGEMENT

MORAD HOSSEINALIFAM
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INDUSTRIEL)
AOÛT 2009

Your file Votre référence
ISBN: 978-0-494-53909-5
Our file Notre référence
ISBN: 978-0-494-53909-5

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

A FRACTIONAL PROGRAMMING APPROACH FOR CHOICE-BASED NETWORK REVENUE MANAGEMENT

présenté par : HOSSEINALIFAM, Morad

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. LANGEVIN André, Ph.D., président

M. SAVARD Gilles, Ph.D., membre et directeur de recherche

M. DESAULNIERS Guy, Ph.D., membre

To my dear parents

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor, Professor Gilles Savard for accepting me in his research group, for his confidence in my abilities and for his encouragements during my work which made difficulties much easier during the last two years.

I would like to thank my parents, my dear brother and sister for their encouragements and support. This work was not possible without their presence. Moreover, I would like to appreciate all my friends for their support.

RÉSUMÉ

Récemment, nous avons vu une augmentation de l'utilisation des modèles des choix du client dans les problèmes de la gestion de revenu. Cet intérêt croissant est principalement dû aux insatisfactions liées aux limitations des modèles traditionnels de la gestion de revenu. Modélisant le comportement du client suivi par des techniques d'optimisation des revenus, qui sont utilisées pour résoudre des problèmes complexes, sont les principales idées à retenir de ces études.

Dans cette recherche, nous considérons le modèle déterministe, de programmation linéaire (CDLP) basé sur les choix de Gallego et al. [20] et les recherches faites par Van Ryzin et Liu [40] et Vulcano [9] dans lesquelles les clients appartiennent à des segments qui se chevauchent. Les prix sont fixes et la firme veut maximiser ses revenus en décidant de l'assortiment optimal de ses offres de produits.

Toutefois, comme un algorithme de génération de colonnes est considéré pour résoudre un CDLP sur un réseau de taille réel, nous faisons face à un sous-problème de programmation linéaire fractionnaire qui est NP-difficile. Nous offrons une approche heuristique simple pour surmonter cette complexité. Selon nos résultats numériques, l'heuristique utilisée, que ce soit en termes de la qualité de la solution obtenue ou du temps de calcul, performe mieux que les approches actuelles.

ABSTRACT

Recently, we have seen an increasing use of customer choice behavior models in revenue management problems. This growing interest is mainly because of dissatisfactions with the limitations of traditional revenue management models. Modeling customer behavior, followed by revenue optimization techniques which are used to deal with such complex models, are main steps in taking advantage of these studies.

In this research, we consider the choice-based, deterministic, linear programming (CDLP) model of Gallego et. al. [20] and further works done by Van Ryzin and Liu [40] and Vulcano [9] in which customers belong to overlapping segments. The prices are fixed and a firm wants to maximize its revenue by deciding the optimal assortment of products to offer.

However, as a column generation algorithm is considered to solve CDLP on real-size network, we face a linear fractional programming subproblem which is NP-hard. We provide a simple heuristic approach to tackle this complexity. According to our numerical results, the heuristic, both in the terms of quality of the obtained solution and processing time, performs better than present approaches.

CONDENSÉ EN FRANÇAIS

De nos jours, la gestion du revenu joue un rôle très significatif dans plusieurs industries. Cette discipline a commencé vers 1972 aux États-Unis dans l'industrie aérienne. Et elle a connu une croissance rapide dès ses débuts [26]. À cette époque, il y avait un besoin croissant de gérer la capacité devant être vendue aux vacanciers avec des billets à bas prix tout en ne perdant pas les revenus provenant des voyageurs d'affaires qui achètent des billets à fort prix mais, généralement, plus tard que les vacanciers.

Ensuite, cette discipline a été étendue à plusieurs autres domaines comme le domaine ferroviaire, de croisière, hôtelier, et ensuite, à d'autres domaines tels que l'énergie, la gestion hospitalière, la vente au détail de la mode, la fabrication, etc. Toutes ces industries essaient d'utiliser les stratégies de la gestion du revenu comme un sous-champ de la recherche opérationnelle pour gérer scientifiquement la demande de leurs produits et services.

L'une des nombreuses définitions académiques pour la gestion du revenu est celle proposée par Cross [12] : " the application of disciplined tactics that predict consumer behavior at the micro market level and optimize product availability and price to maximize revenue growth "; ce qui peut être traduit par : " l'application de tactiques disciplinées qui prédisent le comportement du consommateur au niveau micro-marché et optimisent la disponibilité du produit et le prix afin de maximiser la croissance des revenus ". Plus précisément, nous pouvons dire que la gestion du revenu peut être considérée comme le processus à travers lequel les consommateurs se voient offrir le bon produit à travers les bons canaux de distribution au bon moment et au bon prix de sorte à maximiser les revenus de la firme [36].

Talluri et Van Ryzin [37] définissent la gestion du revenu comme les décisions de gestion de la demande et la méthodologie ainsi que les systèmes requis pour prendre ces décisions. Il y a trois catégories de base pour les décisions de gestion de la demande :

- Décisions structurelles : le format de vente qui est utilisé comme les négociations, les prix affiches ou les enchères; le mécanisme de segmentation ou différentiation à utiliser; et ainsi de suite.
- Décisions du prix : déterminer le prix au fil du temps; décider sur un rabais sur la durée de vie du produit; déterminer les prix affichés, prix individuels; et ainsi de suite.
- Décisions de la quantité : la capacité allouée aux différents segments, produits; accepter ou rejeter une offre; périodes de temps pour offrir ou retenir un produit; et ainsi de suite.

Chaque entreprise peut utiliser une seule ou une combinaison de ces stratégies de gestion du revenu, dépendamment de sa situation. Toutefois, notre travail ne concerne que la troisième catégorie.

La plupart des modèles traditionnels de gestion du revenu sont basés sur une hypothèse de demande indépendante, c'est-à-dire que la "demande est associée avec un produit et est essentiellement indépendante de l'environnement du marché " [37]. Ceci signifie que la demande pour un produit est un nombre fixe et est complètement indépendante de l'environnement compétitif. Par exemple, dans ce type de modèles, il est supposé que les demandes arrivent dans un ordre spécifique dans lequel les demandes à bas prix viennent en premier.

Littlewood [26] présente une approche de résolution pour définir une limite de réservation pour le nombre de sièges qui devraient être assignés aux bas prix dans les réseaux aériens. D'un autre côté, nous pouvons aussi définir la proportion de la capacité qui devrait être réservée pour les passagers qui achètent des billets plus tard à plus grand prix.

L'une des principales limites dans le modèle traditionnel est comment nous devrions mettre en oeuvre les phénomènes de "buy-up " et "buy-down", où "buy-down" est défini comme remplacer un prix plus élèvé par un prix a plus bas lorsque l'entreprise donne un rabais pour un produit et "buy-up" signifie acheter un tarif plus élevé lorsqu'un tarif plus bas n'est pas disponible.

Clairement, la décision d'un consommateur dans un tel environnement pourrait être liée aux rabais et aux prix les plus bas qui lui sont disponibles au moment de la prise de la décision. Toutefois, en réalité, la décision d'un consommateur est non seulement dépendante du prix du produit, mais d'autres facteurs tels que le remboursement, le temps, la date, et les préférences de route qui ont aussi un effet sur son comportement. Plusieurs chercheurs ont proposé quelques stratégies pour prendre en compte ces comportements et surmonter ces sérieuses limites du modèle de la demande traditionnelle.

Cette recherche est principalement motivée par le modèle de la programmation linéaire déterministe basé sur le choix du client (CDLP), proposé par Gallego et al. [20], et son extension avec la considération d'une offre flexible de produits, proposée par Vulcano et. al. [9], où la firme a la flexibilité d'offrir aux clients différents choix afin de rencontrer leur demande; ex. dans les compagnies aériennes, différents choix pour aller d'une même origine à une même destination. Les variables de décision sont la période durant laquelle une firme devrait mettre à la disposition du client un ensemble de produits pour satisfaire sa demande tout en maximisant ses profits. Chaque client appartient à un ou plusieurs segments, qui sont définis comme étant des ensembles de considérations de produits qui se chevauchent.

Les deux principaux défis auxquels nous faisons face dans la mise en oeuvre de la gestion du revenu basée sur le choix sont :

- Modéliser le comportement du choix du consommateur et son estimation à partir des données disponibles.
- Utiliser des méthodes d'optimisation des revenus qui peuvent traiter des modèles complexes de la demande basés sur des choix.

Le meilleur modèle, et le plus répandu, pour étudier comment les consommateurs expriment leurs choix est le modèle " Multinomial Logit" (MNL). Le MNL est une approche paramétrique utilisée pour estimer le comportement du choix du client basée sur différents attributs tels que : le temps, la date de départ, le prix, l'aéroport de départ, etc. En utilisant ce modèle, la probabilité que le consommateur n achète le vol i est donnée par :

$$P_n(i) = \frac{e^{\beta^T x_{in}}}{\sum_{j \in C_n^a} e^{\beta^T x_{jn}} + 1}, \; i \in C_n \tag{1}$$

Avec :

x_{jn} est le vecteur des attributs observables pour l'alternative j disponible au consommateur n au moment de l'achat.

β est le vecteur poids qui devrait être calculé à partir des données.

C_n est l'ensemble des produits offerts au client n.

Pour définir le modèle CDLP, nous considérons un réseau avec m ressources (portions des trajets) qui offre n produits avec $N = \{1, 2, ..., n\}$ l'ensemble des produits et r_j le revenu associé (prix) au produit $j \in N$. Nous étudions la capacité d'usage en définissant le vecteur $c = (c_1, c_2, ..., c_m)$ qui désigne les capacités initiales des ressources (portions des trajets). L'utilisation des ressources selon les produits correspondants est définie par une matrice d'incidence $A = [a_{ij}] \in B^{m \times n}$. Les entrées de la matrice sont définies par:

$$a_{ij} = \begin{cases} 1, & \text{si la ressource } i \text{ est prise par le produit } j, \\ 0, & \text{sinon.} \end{cases}$$

A_j, la j-ième colonne de A, désigne le vecteur d'incidence du produit j et la notation $i \in A_j$ indique que le produit j utilise la ressource i. Il est à noter qu'un produit peut utiliser plus d'une ressource. Le temps est discrétisé et s'écoule sur un nombre fini de périodes T, $t = 1, 2, ..., T$ et il est supposé que nous avons au plus une arrivée par période de temps et chaque client peut acheter un seul produit. La durée de temps est une décision fondamentale. Si nous prenons une unité de temps qui est très petite telle que quelques secondes alors, dans un marché O-D donné, nous allons avoir des réservations juste sur peu de périodes et, sur la plupart d'entre elles nous, n'aurons aucun achat. Vulcano et Van Ryzin [41], en se basant sur des expériences numériques sur les réseaux aériens, suggèrent de diviser la journée à en $T = 140$ petites périodes de temps (considérant approximativement chaque 10 minutes comme une période de temps).

Nous utilisons λ pour désigner la probabilité d'avoir une arrivée dans une période de temps et nous divisons les clients en L segments différents. Un ensemble de considérations $C_l \subseteq N$, $l = 1, 2, ..., L$ est utilisé pour décrire chaque segment. Ici, la différence entre notre modèle et les travaux précédents, modèles basés sur le choix du client, est plus claire. Gallego et al. [20] considèrent un seul segment $C_1 = N$ et contrairement à Van Ryzin et Liu [40], nous pouvons avoir des segments qui se chevauchent: $C_l \bigcap C_{l'} \neq \emptyset$ pour certans $l \neq l'$.

Si nous avons une arrivée, p_l représente la probabilité qu'un client qui arrive appartienne au segment l avec $\sum_{l=1}^{L} p_l = 1$. Nous considérons un processus de Poisson pour les flux d'arrivée des clients du segment l avec un taux $\lambda_l = \lambda p_l$ pour un taux d'arrivée total de

$\lambda = \sum_{l=1}^{L} \lambda_l$.

Dans chaque période de temps t, la firme doit décider de son offre (*i.e.* un sous-ensemble de produits $S \subset N$, que la firme rend disponible aux clients). Si un ensemble S est offert, la quantité déterministe $P_j(S)$ indique la probabilité de choisir un produit $j \in S$, et $P_j(S) = 0$ si $j \notin S$. Par la loi de la probabilité totale, nous avons $\sum_{j \in S} P_j(S) + P_0(S) = 1$, où $P_0(S)$ est la probabilité de non achat.

Comme il a déjà été déclaré, nous utilisons un modèle " Multinomial Logit " (MNL) afin de trouver les probabilités du choix du client. Selon le choix du modèle MNL, le vecteur $v_l \geq 0$ est un vecteur préférence du client pour les produits disponibles dans l'ensemble des considérations C_l et v_{l0} représente la préférence de non achat. $P_{lj}(S)$ désigne la probabilité de vendre le produit $j \in C_l \bigcap S$ à un client du segment l quand l'ensemble S est offert. Donc, la probablité du choix du client peut être exprimée comme suit:

$$P_{lj}(S) = \frac{v_{lj}}{\sum_{h \in C_l \bigcap S} v_{lh} + v_{l0}}. \tag{2}$$

Il peut être obtenu, à partir de l'équation (2), que $P_{lj}(S) = 0$ si $v_{lj} = 0$ qui peut être un résultat de $j \notin C_l$ ou $j \notin C_l \bigcap S$. Nous suppos $v_{l0} > 0$ pour tous les segments $l = 1, 2, ..., L$. Dans le cas le plus général, puisque que la firme ne connaît pas le segment correspondant à un client donné, nous considérons $P_j(S)$, la probabilité que cette dernière vende le produit j à un client quelconque qui arrive, comme:

$$P_j(S) = \sum_{l=1}^{L} p_l P_{lj}(S). \tag{3}$$

Le revenu attendu, en offrant l'ensemble $S \subset N$, d'un client qui arrive est donné par:

$$R(S) = \sum_{j \in S} r_j P_j(S). \tag{4}$$

Puisque nous offrons l'ensemble S, soit $P(S) = (P_1(S), ..., P_n(S))^{\top}$ le vecteur des probabilités d'achat et A la matrice d'incidence des ressources utilisées par les produits. Donc le vecteur des probabilités de la capacité de consommation $Q(S)$ est donné par:

$$Q(S) = AP(S), \tag{5}$$

où $Q(S) = (Q_1(S), ..., Q_m(S))^{\top}$ et $Q_i(S)$ indique la probabilité d'utiliser une unité de capacité de la portion de trajet $i, i = 1, 2, ..., m$.

La décision de la firme consiste à déterminer à n'importe quelle période de temps t, quel ensemble de produits devrait être offert, tout en ignorant le segment du client concerné. Toutefois, puisque les probabilités des choix sont homogènes dans le temps et que la demande est une variable déterministe, il importe seulement combien de fois chaque ensemble S est offert, savoir durant quelles périodes exactes il est offert n'est pas important. Une autre hypothèse est que nous permettons à la variable $t(S)$ d'être continue (*i.e.* la firme peut offrir un ensemble S pour une période de temps complète ou une fraction de ce temps).

L'objectif de ce modèle est de maximiser le revenu de la firme en décidant du nombre de périodes de temps où chaque ensemble de produits sera offert. En se référant à l'équation (3) dans Vulcano et al. [9], ceci mène au programme linéaire (LP) suivant :

$$\begin{aligned} V^{CDLP} = \quad & \max \sum_{S \subset N} \lambda R(S) t(S) \qquad (6) \\ \text{sujet à} \quad & \sum_{S \subset N} \lambda Q(S) t(S) \leq c, \\ & \sum_{S \subset N} t(S) \leq T, \\ & t(S) \geq 0, \forall S \subset N. \end{aligned}$$

Il y a $m + 1$ contraintes dans le problème (6), où les premières m contraintes sont relatives à la disponibilité de la capacité et la dernière contrainte est pour la disponibilité du temps. En raison du nombre de contraintes ($m + 1$), nous pouvons avoir un maximum de $m + 1$ variables positives dans la base. Il y a quelques remarques qui devraient être mentionnées ici à propos du modèle CDLP et de sa solution optimale.

Premièrement, nous devons choisir comment appliquer la solution du modèle CDLP dans notre problème réel et assigner un temps de début et de fin pour l'offre de chaque produit. Comme il a été mentionné avant, la solution du modèle CDLP ne nous donne pas une séquence de produits ou de temps. Toutefois, pour ordonner l'ensemble d'offres, plusieurs approches heuristiques peuvent nous aider. Van Ryzin et Liu [40] ont développé une décomposition heuristique efficiente pour surmonter ce problème.

Deuxièmement, dans le problème (6), il y a un nombre exponentiel de variables primales. Ceci signifie qu'un problème avec n produits a $2^n - 1$ sous-ensembles possibles non-vides de produits. Malgré le grand nombre de variables pour des problèmes pratiques de la réalité, qui font en sorte qu'il est impossible d'énumérer tous les ensembles d'offres, il y a au plus $m + 1$ contraintes. Ceci mène à l'idée d'utiliser des techniques de génération de colonnes afin de résoudre les problèmes pratiques de la réalité.

Gallego et al. [20] suggèrent l'utilisation des techniques de génération de colonnes pour résoudre les modèles CDLP réels. Les étapes de cet algorithme sont :

- Étape 1: Commencer par résoudre un LP réduit; i.e. considérer seulement un nombre limité de colonnes (sous-ensembles) au lieu de toutes les énumérer.
- Étape 2: Construire un sous-problème en utilisant la solution duale du LP réduit pour trouver une colonne avec un coût réduit le plus positif.
- Étape 3: Ajouter la colonne avec le coût réduit positif au LP réduit et le résoudre.

- Étape 4: S'il y a aucune colonne d'entrée avec un coût réduit positif, alors la solution courante est optimale.

Retournons au modèle CDLP original (6), mais juste avec un nombre de colonnes initiales limité indiquées par $\mathcal{N} = \{S_1, S_2, ..., S_k\}$. Ceci nous amène au modèle CDLP réduit suivant :

$$\begin{aligned} V^{CDLP-R} \quad &= \quad \max \sum_{S \in \mathcal{N}} \lambda R(S) t(S) && (7) \\ \text{sujet à} \quad & \sum_{S \in \mathcal{N}} \lambda Q(S) t(S) \leq c, \qquad (\pi) \\ & \sum_{S \in \mathcal{N}} t(S) \leq T, \qquad (\sigma) \\ & t(S) \geq 0, \forall S \in \mathcal{N}. \end{aligned}$$

Soient $\pi \in \mathbf{R}^m$ correspondant aux prix duaux des m premières contraintes de la capacité et $\sigma \in \mathbf{R}$, le prix dual, correspondant à la contrainte unidimensionnelle du temps. Maintenant, pour la prochaine étape dans l'algorithme de génération de colonnes, nous construisons un sous-problème de génération de colonnes pour trouver la prochaine colonne à ajouter à notre ensemble $\mathcal{N}$, qui a un coût réduit le plus positif et qui n'est pas encore incluse. Cette colonne est obtenue en résolvant le sous-problème suivant:

$$\max_{S \subseteq N} \{\lambda R(S) - \lambda \pi^\top Q(S) - \sigma\} = \max_{S \subseteq N} \{\lambda R(S) - \lambda \pi^\top Q(S)\} - \sigma. \qquad (8)$$

Ensuite, pour expliciter la formulation (8), un vecteur binaire $y \in \mathbf{B}^n$ est défini comme suit. Supposons qu'un ensemble S est offert maintenant, alors nous désignons :

$$y_j = \begin{cases} 1, & \text{si } j \in S, \\ 0, & \text{sinon.} \end{cases}$$

Après avoir introduit les variables binaires y_j, la formulation (8) peut être exprimée comme suit :

$$\max_{y\in\{0,1\}^n}\left\{\sum_{l=1}^{L}\lambda_l\frac{\sum_{j\in C_l}(r_j - A_j^\top\pi)v_{lj}y_j}{\sum_{i\in C_l}v_{li}y_i + v_{l0}}\right\} - \sigma, \tag{9}$$

Ou, d'une manière équivalente,

$$\max_{y\in\{0,1\}^n}\left\{\sum_{j=1}^{n}\left(r_j - A_j^\top\pi\right)y_j\left(\sum_{l=1}^{L}\frac{\lambda_l v_{lj}}{\sum_{i\in C_l}v_{li}y_i + v_{l0}}\right)\right\} - \sigma. \tag{10}$$

Notez que nous suppos $v_l \geq 0$ pour être certain que notre dénominateur est plus grand que zéro en tout temps. Si la valeur optimale du problème (10) est positive, alors la solution optimale pour le problème (10) est la prochaine colonne entrante du CDLP réduit (7). Alors, nous mettons à jour le CDLP réduit (7) avec une nouvelle colonne et on réitère. Finalement, s'il n'y a pas de solution pour le problème (10) avec une valeur objectif positive, alors la solution actuelle pour le problème CDLP maître (6) est optimale.

Le problème (10) est appelé problème de programmation fractionnaire, dans lequel on cherche à maximiser la somme de plusieurs ratios. Vulcano et al. [9] prouvent que le problème du sommet minimum, qui est un problème NP-difficile, peut être réduit au problème (10); par conséquent le problème (10) est un problème NP-difficile [Théorème 1,[9]].

Il y a plusieurs approches de résolution pour le problème (10). Une façon de résoudre le sous-problème de génération des colonnes est de le reformuler comme un problème mixte en numbres entiers [9]. Nous considérons le problème (10). Nous commençons par définir de nouvelles variables x_l, $l = 1, ..., L$ comme suit :

$$x_l = \frac{1}{\sum_{i\in C_l}v_{li}y_i + v_{l0}}. \tag{11}$$

Ensuite, la substitution des x_l dans la formulation (10) mène à la formulation suivante:

$$\max \sum_{l=1}^{L} \sum_{j \in C_l} \lambda_l (r_j - A_j^{\top} \pi) v_{lj} y_j x_l \tag{12}$$

$$\begin{aligned} \text{sujet à} \quad & x_l v_{l0} + \sum_{i \in C_l} v_{li} y_i x_l = 1, \quad && l = 1, ..., L, && (13) \\ & y_j \in \{0, 1\}, && j \in N, && (14) \\ & x_l \geq 0, && l = 1, ..., L. && (15) \end{aligned}$$

Nous pouvons voir que les termes non-linéaires $y_i x_l$ apparaissent dans (12) et (13). Ces termes peuvent être linéarisés en utilisant le théorème proposé par Wu [43]: un terme polynomial mixte 0-1 $z = xy$, où x est une variable continue et y est une variable binaire, peut être représentée par le système linéaire suivante :

$$\begin{aligned} x - z &\leq K - Ky, && (16) \\ z &\leq Ky, && (17) \\ z &\leq x, && (18) \\ z &\geq 0. && (19) \end{aligned}$$

Soit $z_{li} = x_l y_i$. Avec les nouvelles variables, la formulation (10) peut être réécrite comme suit :

$$\begin{aligned} & \max \sum_{l=1}^{L} \sum_{j \in C_l} \lambda_l (r_j - A_j^{\top} \pi) v_{lj} z_{lj} \\ & \text{sujet à} \quad x_l v_{l0} + \sum_{i \in C_l} v_{li} z_{li} = 1, && \forall l, l = 1, ..., L \qquad (20) \\ & \qquad x_l - z_{li} \leq K - K y_i, && \forall l, l = 1, ..., L, i \in C_l, \\ & \qquad z_{li} \leq x_l, && \forall l, l = 1, ..., L, i \in C_l, \\ & \qquad z_{li} \leq K y_i, && \forall l, l = 1, ..., L, i \in C_l, \\ & y_j \in \{0, 1\}, \ x_l \geq 0, \quad z_{li} \geq 0. \end{aligned}$$

K doit être un nombre plus grand que x. Comme nous avons défini $x_l = \frac{1}{\sum_{i \in C_l} v_{li} y_i + v_{l0}}$ et y_j ne prend que des valeurs binaires, il est suffisant de prendre $K \geq \frac{1}{\underline{v}}$ où $\underline{v} = \min\{v_{li} : i = 0, 1, ..., n; l = 1, 2, ..., L\}$.

Le fait que le sous-problème de génération de colonnes est un problème d'optimisation NP-difficile nous force à utiliser une approche alternative pour pouvoir implémenter cet algorithme pour des problèmes pratiques. Vulcano et al. [9] ont proposé une heuristique "gloutonne" basée sur celle proposée par Prokopyev [30] avec une complexité $O(n^2 L)$ pour faire résoudre ce même problème.

Cette heuristique commence par un ensemble vide S, et tout en prenant en compte la contribution marginale maximale de la solution actuelle, ajoute progressivement des nouveaux produits à l'ensemble actuel S.
L'algorithme est présenté dans les étapes suivantes :

- Étape 1 : Pour tout produit j tel que $r_j - A_j^\top \pi \leq 0$, poser $y_j = 0$.
- Étape 2 : Soit $S' \subset N$ l'ensemble de produits j avec aucune valeur assignée à y_j.
- Étape 3 : Calculer $j_1^* = \text{argmax}_{j \in S'} \left\{ \sum_{l=1}^{L} \frac{(r_j - A_j^\top \pi) v_{lj}}{v_{lj} + v_{l0}} \right\}$. Poser $S := \{j_1^*\}, S' := S' - \{j_1^*\}$.
- Étape 4: **faire tant que** S est modifié
  - Calculer $j^* := \text{argmax}_{j \in S'} \left\{ \sum_{l=1}^{L} \lambda_l \frac{\sum_{i \in C_l \cap (S \cup \{j\})} (r_i - A_i^\top \pi) v_{li}}{\sum_{i \in C_l \cap (S \cup \{j\})} v_{li} + v_{l0}} \right\}$.
  - Si Valeur $(S \bigcup \{j^*\}) >$ Valeur (S), alors $S := S \bigcup \{j^*\}$, et $S' := S' - \{j^*\}$.

  Fin tant que.
- Étape 5: Pour tout $j \in S$, poser $y_j = 1$. Pour tout $j \notin S$, poser $y_j = 0$.

Le temps et la qualité de la solution peuvent être considérés comme deux principaux concepts clés pour montrer l'efficience d'un algorithme. Nous continuons en présentant une nouvelle approche heuristique pour améliorer l'efficience de la méthode actuelle. On considère le problème de somme de ratios suivant :

$$\max_x \sum_{l=1}^{L} Q_l(x) = \sum_{l=1}^{L} \frac{P_l(x)}{D_l(x)} = \sum_{l=1}^{L} \frac{\sum_{j=1}^{m} p_{lj} x_j}{\sum_{j=1}^{m} d_{lj} x_j}, \tag{21}$$

où $D_l(x) > 0, \forall l = 1, ..., L$.

Almogy et Levin [1] ont essayé de maximiser le problème de la somme des ratios en transformant le problème (21) en un problème paramétrique équivalent. Néanmoins, Falk et Palocsay [17], par un exemple numérique, ont démontré que cet algorithme ne fonctionne pas en général. En utilisant l'idée de l'algorithme de Almogy et Levin et Dinkelbach [3], nous développons une heuristique pour simplifier les étapes de calculs dans l'algorithme de génération des colonnes en un temps polynomial.

Pour ce faire, en considérant le problème (21), nous définissons le problème paramétrique associé ($F(\rho)$) comme suit:

$$F^{(k-1)}(\rho) = \max_x \left\{ \sum_{l=1}^{L} \frac{P_l(x) - \rho_l D_l(x)}{D_l(x^{k-1})} \right\}, \tag{22}$$

qui est une fonction convexe non-croissante de ρ [32]. ρ_l est défini comme suit

$$\rho_l = \frac{P_l(x^{(k)})}{D_l(x^{(k)})}, \tag{23}$$

où $x^{(k)}$ est la solution du problème de maximisation (22) dans l'itération précédente de l'algorithme et, dans la première itération, on commence par $x^{(0)}$, une solution faisable. En prenant en compte la formulation paramétrique $F(\rho)$, les étapes de l'algorithme sont décrites comme :

- Étape 1: Prendre $x^{(0)} \in S$, calculer $\rho_l^{(1)} = \frac{P_l(x^{(0)})}{D_l(x^{(0)})}$, $\forall l = 1, ..., L$; et poser $k := 1$.
- Étape 2: Déterminer $x^{(k)} = \text{argmax}_{x \in s} \left\{ \sum_{l=1}^{L} \frac{P_l(x) - \rho_l^{(k)} D_l(x)}{D_l(x^{k-1})} \right\}$.
- Étape 3: Si $F^{(k)}(\rho^{(k)}) = 0$ alors $x^* = x^{(k)}$ est notre solution, Stop.
- Étape 4: Poser $\rho_l^{(k+1)} = \frac{P_l(x^{(k)})}{D_l(x^{(k)})}$, $\forall l = 1, ..., L$; poser $k := k + 1$; aller à l'étape 2.

Où k désigne le nombre d'itérations et S est l'ensemble des solutions réalisables. Pour évaluer l'algorithme exact et les deux autres approches heuristiques, nous considérons deux exemples pour les modèles de réseau de la gestion du revenu basée sur les choix avec des segments qui se chevauchent dans lesquels les clients choisissent leurs produits basés sur un modèle de choix "Multinomial Logit". Ensuite, nous mettons en oeuvre différentes stratégies développées dans ce travail et nous rapportons les résultats numériques. En prenant en compte les résultats de calculs, nous évaluons différentes approches. Les résultats démontrent que l'agorithme exact n'est pas applicable pour les problèmes pratiques de la réalité.

Cependant, même s'il n'y a aucune garantie qu'une solution optimale pourrait être trouvée avec l'heuristique basée sur la méthode de Dinkelbach, selon nos résultats de calculs, cette heuristique a une performance remarquable en termes de la qualité de la solution obtenue et du temps de clacul: elle performe mieux que l'heuristique gloutonne.

Il y a plusieurs sujets qui méritent d'être considérés dans les prochaines recherches. Un serait d'améliorer l'algorithme de génération des colonnes de telle manière qu'au lieu d'une seule colonne, nous aurions plusieurs colonnes entrantes dans le problème maître qui permettent d'améliorer l'efficience de l'algorithme. Simultanément, en vue d'atteindre une plus grande performance, améliorer les approches heuristiques devrait être utile également.

Une autre extension intéressante de l'algorithme pourrait être de supposer des vecteurs incertains de préférence pour les clients. Ceci signifie que la probabilité de choisir un certain produit pourrait changer durant l'horizaon de la réservation et nous mener à avoir une meilleure interprétation du comportement du consommateur et ainsi, des meilleures politiques de prises de décisions. De plus, étudier les approches de décomposition de la programmation dynamique pourraient être une approche intéressante pour améliorer les procédures disponibles.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Revenue management nowadays plays a very significant role in a wide range of industries. It started to grow rapidly from its beginnings in the United States airline industries around 1972 [26]. At that time, there was a growing need to manage the capacity that should be sold to leisure travelers with low fare tickets, while the firms did not want to lose their revenue from business travelers who buy high-fare tickets but generally purchase later than leisure travelers.

Afterwards, this field was extended to many other domains such as railways, cruises, hotels, and moreover, other areas like energy, hospitality, fashion retail, manufacturing, etc. All of these industries try to use revenue management strategies as a subfield of operations research to scientifically manage demand for their products and services.

One of the most academic definitions for revenue management is proposed by Cross [12] as " the application of a disciplined tactics that predict consumer behavior at the micro market level and optimize product availability and price to maximize revenue growth." More concisely, we can say revenue management can be considered to be the process through which the right customers are offered the right product through the right distribution channels at the right time and at the right price such that the revenue of a firm are maximized. [36]

Talluri and Van Ryzin [37] define revenue management as demand management decisions and the methodology and systems required to make them. There are three basic categories for demand-management decisions:

- Structural decisions: the selling format that is used such as negotiations, posted prices or auctions; the segmentation or differentiation mechanism to use; and so on are studied in this category.

- Price decisions: setting the price over time; deciding on a discount over the lifetime of the product; setting the posted prices, individual prices; and so on are studied in this category.

- Quantity decisions: the capacity allocated to different segments, products; accepting or rejecting an offer; periods of time to offer or withhold a product; and so on are studied in this category.

Any given business, depending on their current situation, may use one or a combination of structural, price or quantity-based revenue management strategies. However, we could classify this research in the third category.

Traditional revenue management systems were based on independent demand assumption, where they assume that demand for a given product is essentially independent of the market environment [39]; on the other hand, demand is not affected by the competitive environment, such as possible product alternatives, offered by same firm or other competitors. However, studying customer buying behavior denotes that this assumption is not true in reality and bookings are the function of available fare products, discounts, etc. Hence considering such behaviors while making control decisions could have a reasonable improvement in the obtained revenue by the firm.

In this research, we study a choice-based network revenue management, and to do so, we describe two main challenges in customer choice behavior models. The first is modeling how a customer makes his decisions in any period of time and estimating parameters which could describe such behaviors. The second is to employ all of the gathered information as a linear programming model in a revenue management system that could efficiently help a firm choose its selling policies.

This research is mainly motivated by the customer choice-based deterministic linear programming (CDLP) model, proposed by Gallego et al. [20] and its extension with considering a flexible product offering proposed by Vulcano et. al. [9], where the firm has the flexibility to offer the customers different choices to serve their demand; *e.g.* in airline companies, different choices to go from the same origin to the same destination. The decision variables are the length of time during which a firm should make available a set of products to satisfy customers' demand while it wants to maximize its profit as well. Each customer belongs to one or more segments, which are defined by overlapping consideration sets of products.

This CDLP model can be solved directly in very small instances, but for real world problems with a large number of variables, we are obliged to use special techniques in solving large scale optimization problems. The column generation algorithm, as one of the most well known techniques in such problems, is used to solve practical CDLP models.

The main challenge of this approach is the fact that the column generation algorithm's subproblem in our case is a special case of a linear fractional programming problem, where we want to maximize a sum of several ratios. In the next chapter, we see that this problem is proved to be NP-hard, so we should look after efficient ways to face this challenge.

The contribution of this dissertation to revenue management is to introduce an efficient algorithm to solve choice-based network revenue management models. Our results show that either in the terms of quality of the obtained solution and processing time this algorithm performs much better than present approaches.

In the next chapter, we present a review on choice-based revenue management networks, followed by a review on some theory and conventional methods used in linear fractional programming models. In the third chapter, we consider the CDLP model for a firm that faces streams of customers from overlapping segments while it needs to decide which alternatives to offer at any period of time to maximize revenue and satisfy demands. We investigate available solution approaches to solve the column generation algorithm's subproblem, which is known to be NP-hard. Finally, we provide a heuristic with high quality results to tackle this complexity.

In chapter four, considering two examples, we implement different strategies studied in the previous chapter. Taking into account the computational results, we evaluate different solution approaches based on the quality of the obtained results and the computational time consumed for the operations. Finally, we conclude this research and discuss the perspective of this project in chapter five.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, we first present a summary of choice-based revenue management, afterwards we will continue with some theoretical aspects and applications of linear fractional programming, followed by some solution methods and examples.

2.2 Choice-based revenue management

Talluri and Van Ryzin [37] define revenue management as demand management decisions and the methodology and systems required to make them. Revenue management models began to be studied academically by Littlewood [26] by presenting some simple techniques to solve traditional revenue management models.

Most traditional revenue management models are based on independent demand assumption which undertakes that "demand is associated with a product and is essentially independent of the market environment" [37]. This means that demand for a given product is a fixed number and it is completely independent of the competitive environment. For example, in these kinds of models, it is assumed that demands arrive in a specific order in which low fare demand comes first.

Littlewood [26] presents a solution approach to set a booking limit for the number of seats which should be assigned to low fares in airline networks. He assumes a model with two high and low fare classes with the total capacity C. Demand for class two denoted by D_2 arrives before demand for class one while associated price for class one denoted by r_1 is strictly larger than the associated price for class two. Distribution for each class j, $j = 1, 2$ is denoted by $F_j(.)$ and the problem is to find how much of the capacity should be reserved for passengers who purchase tickets later with higher fare demand.

A simple marginal analysis is used to find the optimal solution. Suppose a customer from class two arrives while we have x units remaining capacity. As we know that customers from class one arrive later than class two, if we do not accept this request, we will lose revenue r_2 while we can sell this marginal unit to customers from class one if and only if demand for this class exceeds x (i.e. $D_1 \geq x$) and we have the expected marginal value $r_1 P(D_1 \geq x)$. Hence, it is rational to accept class two customers till their revenue exceeds this marginal value, or in the other hand, if and only if

$$r_2 \geq r_1 P(D_1 \geq x). \tag{2.1}$$

As the right-hand side of (2.1) is decreasing in x, a protection level, y_1^*, can be defined such that we do not accept anymore class two customers if the remaining capacity is y_1^* or less. This means that the following equations hold for y_1^*

$$r_2 < r_1 P(D_1 \geq y_1^*) \quad \text{and} \quad r_2 \geq r_1 P(D_1 \geq y_1^* + 1). \tag{2.2}$$

The optimal protection level, y_1^* by considering a continuous distribution $F_1(x)$ can be obtained by

$$r_2 = r_1 P(D_1 > y_1^*) \quad \text{or equivalently} \quad y_1^* = F_1^{-1}(1 - \frac{r_2}{r_1}). \tag{2.3}$$

This equation (2.3) was knows as Littlewood's rule.

One of the main limitations in the traditional models is how we should implement buy-up and buy-down phenomenons, where buy-down is replacing a lower fare for a higher fare when the firm gives a discount for a product, and buy-up means buying a higher fare when a low fare is not available.

Clearly, a customer's decision in such an environment could be related to the discounts and the lowest price, which is available to customers needing products at the time of making a decision. However, in reality, a customer's decision is not only dependent on the price of the product, but other factors like refundability, time, date and path preferences also have an effect on their behavior. Many researchers have proposed some strategies to take into account such behaviors and overcome these serious limitations in traditional demand assumptions.

Figure (2.1) shows the position of a revenue management model in the marketing system and its internal connections between the forecaster and optimizer and as a revenue management cell its output and input data to the whole system. Customers' purchase history, product and pricing information are a collection of data which are used to estimate and forecast customer behavior. Based on forecasting and estimations, the firm decides its optimal policy and by taking into account allocation and overbooking controls, offers its products to customers.

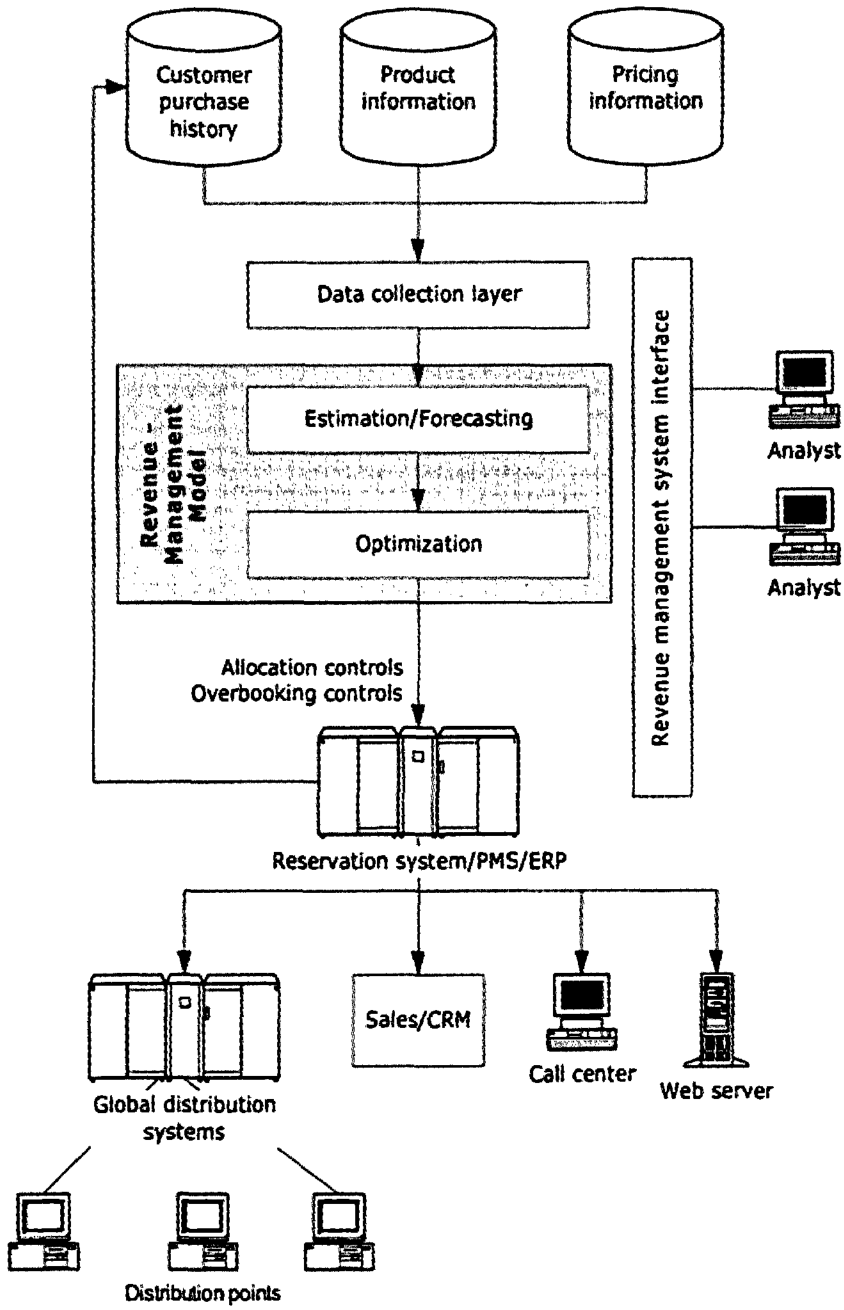

Figure 2.1 Information process flow in a revenue management system [37]

Belobaba [5] introduces an expected marginal seat revenue (EMSR) heuristic to implement buy-up behavior in traditional models, and in the follow-up work done by Belobaba and Hopperstad [4] they show meaningful importance of considering customer choice decision behavior as well. They studied a passenger purchase behavior simulation system by including passengers' preferences in airline, time, date, path and price sensitivity.

Talluri and Van Ryzin [38] provide a complete characterization of an optimal policy under a general discrete choice model of customer behavior in a single leg revenue management model. They propose the fact that an optimal policy is made up of selecting a set of efficient offer sets, where these sets are a sequence of nondominated sets providing the highest positive exchange between expected capacity assumption and expected revenue.

Gallego et al. [20] provide a customer choice-based LP model for network revenue management. They suppose that with a flexible product offering, the firm has the ability to provide customers alternative products to serve the same market's demands. One limitation of their market demand model is that it does not allow any kind of segmentation.

Van Ryzin and Liu [40] use the analysis of the model provided by Gallego et al. to extend the concept of efficient sets. They prove that when capacity and demand are scaled up proportionately, revenue obtained under choice-based deterministic linear programming converges to the optimal revenue under the exact formulation. They present a market segmentation model to describe choice behavior. The segments are defined by disjoint consideration sets of products, where a consideration set is a subset of the products provided by the firm which customers view as options.

Two main challenges that we face in implementing a choice-based revenue management are:

- Modeling customer choice behavior and its estimation from available data.
- Using revenue optimization methods that can deal with complex, choice-based models of demand.

To model customer choice behavior we can assume that each customer wants to maximize his utility while his utility for alternatives is a random variable. The firm is offering a set of alternatives $C = \{1, 2, ..., m\}$ for the customer n where he has a choice (or consideration) set $C_n \subset C$ with the utility U_{in} for each alternative $i \in C_n$. This utility without loss of generality can be decomposed into two deterministic (also called expected utility) denoted v_{in} and a mean-zero random component ε_{in}. Hence, we have utility function as follows:

$$U_{in} = v_{in} + \varepsilon_{in}. \tag{2.4}$$

In many cases, the representative component v_{in} is modeled as a linear combination of several attributes,

$$v_{in} = \beta^{\top} x_{in}, \tag{2.5}$$

where β is an unknown vector of weights that should be computed from data and x_{in} is the vector of observable attributes for alternative i available to customer n at time of purchase, such as time and date of departure, price, departure airport, airline brand, etc.

Let denote the no-purchase alternative as "0", then the probability that customer n chooses alternative j from the set $C_n \cup \{0\}$ can be expressed as

$$P_n(i) = \mathbb{P}(U_{in} \geq U_{jn}, \forall j \in C_n \cup \{0\}). \tag{2.6}$$

One of the best and most commonly used models to study how customers make their choices is the multinomial logit (MNL) model [6]. In this model it is assumed that the ε_{in}s in the utility functions are independent and identically-distributed random variables with a Gumbel distribution having cumulative distribution function

$$F(x) = \mathbb{P}(\varepsilon_{in} \leq x) = \exp(-\exp(-\mu(x - \eta))), \tag{2.7}$$

where μ is a positive scale parameter and η is a location parameter. The probability that customer n chooses alternative $i \in C_n$ in an MNL model is given by

$$P_n(i) = \frac{e^{\mu v_{in}}}{\sum_{j \in C_n^a} e^{\mu v_{jn}} + 1}, \tag{2.8}$$

where the one in the denominator represents the no-purchase utility (i.e. $v_{0n} = 0$ causes to have $e^{\mu v_{0n}}$=1). Now if we model the representative component v_{in} as a linear combination of several attributes, μ can not be distinguished from the overall scale of β and generally it is assumed to be 1. So the formulation (2.8) in the case of linear-in-parameters utilities can be represented as

$$P_n(i) = \frac{e^{\beta^\top x_{in}}}{\sum_{j \in C_n^a} e^{\beta^\top x_{jn}} + 1}, \; i \in C_n. \tag{2.9}$$

Vulcano et al. [9] consider the CDLP model of Gallego et al. [20] and further works done by Van Ryzin and Liu [40]. They extended the model to a more general case, where customers can belong to more than one segment according to a MNL model.

Regarding the large number of variables in a real-size network, they develop a column generation algorithm to solve this CDLP model. However, the subproblem of the column generation algorithm is formulated as a 0-1 fractional programming problem where the sum of several ratios should be maximized. Because of the NP-hardness of this problem, they propose implementing a greedy heuristic algorithm to solve the subproblem in polynomial time.

Inspired by the results of Vulcano et al. [9], we consider the CDLP formulation, the column generation algorithm and its subproblem, and we present a new heuristic method with better efficiency to overcome the complexity of the fractional linear programming subproblem.

In the next section, we present an introduction to the theory and applications of Linear Fractional Programming (LFP) and its relationship with Linear Programming (LP). Some solution methods and examples are presented for LFP problems.

2.3 Linear Fractional Programming - General Form

A linear fractional programming problem is formulated as

$$\max_{x \in S} \sum_{l=1}^{L} Q_l(x) = \sum_{l=1}^{L} \frac{P_l(x)}{D_l(x)} \tag{2.10}$$

where $P_l(x)$ and $D_l(x)$ are affine functions with $D_l(x) > 0 \, \forall l$, $l = 1, 2, .., L$ and $\forall x \in S$, and S is a set of feasible solutions.

Depending on the number of ratios L, whether $L = 1$ or $L \geq 2$, we are facing the problem of maximization of a single ratio or a sum of several ratios.

2.3.1 Linear Fractional Programming - Single Ratio

A single ratio LFP problem was first introduced by Bela Martos in 1964 [27]. An LFP, which is called a hyperbolic programming problem with a single ratio is formulated as follows:

$$\begin{aligned} \max_{x} \quad & Q(x) = \frac{P(x)}{D(x)} = \frac{\sum_{j=1}^{n} p_j x_j + p_0}{\sum_{j=1}^{n} d_j x_j + d_0} \\ \text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, 2, \ldots, m, \\ & x_j \geq 0, \quad j = 1, 2, \ldots, n, \end{aligned} \tag{2.11}$$

where $D(x) > 0, \forall x \in S$.

LFP problems generally deal with the efficiency and effectiveness concepts. They take into account the maximization of the firm's efficiency. We define the efficiency as the ratio of a firm's profit on the labor and production costs. Nowadays, because of a deficit of natural resources, the use of the optimization models becomes more and more applicable. Therefore, LFP as a modelization tool is applied to tackle real-world problems related to the optimization of efficiency.

For example, suppose that a manufacturer is producing five types of product, A, B, C, D and E. The manufacturer has an order from its customers to produce 100, 150, and 300 units of the products B, C and D, respectively, and 150 units without type detailing. The manufacturer wishes to formulate a production plan that maximizes its profit gained per unit of cost with respect to resource availability. All related data are depicted in Table (2.1).

The maximum resource availability for material 1 and material 2 are 500 and 700, respectively. The manufacturer is interested in satisfying its orders while taking into con-

Table 2.1 Manufacturer's resource need

	A	B	C	D	E
Material 1	20	10	-	-	-
Material 2	-	-	22	21	26
Price \$/unit	400	360	395	330	400
Cost \$/unit	300	260	300	290	350

sideration the desire to obtain maximum efficiency.

Let $x_j, j = 1, 2, ..., 5$ denote the number of unknown quantities of products A, B, C, D and E. In this case, we can formulate this problem as follows:

$$\max_x Q(x) = \frac{P(x)}{D(x)} = \frac{400x_1 + 360x_2 + 395x_3 + 330x_4 + 400x_5}{300x_1 + 260x_2 + 300x_3 + 290x_4 + 350x_5} \tag{2.12}$$

subject to:

$$\begin{aligned} 20x_1 + 10x_2 &\leq 500, \text{(Resource availability for material 1)}, \\ 22x_3 + 21x_4 + 26x_5 &\leq 700, \text{(Resource availability for material 2)}, \\ x_2 \geq 100, x_3 \geq 150, x_4 &\geq 300, \text{(Demands satisfaction of the clients)}, \\ x_1 + x_2 + x_3 + x_4 + x_5 &\geq 700, \text{(Whole demand satisfaction)}, \\ x_j \geq 0, j = 1, 2, ..., 5. \end{aligned}$$

Relationship between LFP and LP

The following considerations show how and when an LFP problem can be reformulated as an LP problem.

1. In the objective function of the LFP problem (2.11), if all $d_j = 0, (j = 1, 2, ..., n)$ and $d_0 \neq 0$ then this objective function changes to the following form which is a linear function.
$$Q(x) = \sum_{j=1}^{n} \frac{p_j}{d_0} x_j + \frac{p_0}{d_0}.$$

2. For the case of $p_j = 0, (j = 1, 2, ..., n)$, the objective function of the LFP problem (2.11) changes to:
$$Q(x) = \frac{P(x)}{D(x)} = \frac{p_0}{\sum_{j=1}^{n} d_j x_j + d_0}$$
and maybe replaced with function $D(x)$. In this case maximization of the original objective function $Q(x)$ must be substituted with minimization of a new objective function $D(x)$ on the same feasible set S.

3. Finally, for the case where vectors $p = (p_1, p_2, ..., p_n)$ and $d = (d_1, d_2, ..., d_n)$ are linearly dependant, there exists $\mu \neq 0$ such that $p = \mu d$ and the objective function of the LFP problem (2.11) changes to
$$Q(x) = \frac{P(x)}{D(x)} = \frac{\sum_{j=1}^{n} \mu d_j x_j + p_0}{\sum_{j=1}^{n} d_j x_j + d_0} = ... = \mu + \frac{p_0 - \mu d_0}{\sum_{j=1}^{n} d_j x_j + d_0}$$
and maybe replaced with function $D(x)$. Based on the sign of the term $\{p_0 - \mu d_0\}$, for the positive (negative) sign of this term, the maximization of the original objective function $Q(x)$ must be substituted with minimization (maximization) of a new objective function $D(x)$ on the same feasible set S.

Graphical method to solve LFP problems

Let consider a two variable LFP problem:

$$\max_{x} \quad Q(x) = \frac{P(x)}{D(x)} = \frac{p_1x_1 + p_2x_2 + p_0}{d_1x_1 + d_2x_2 + d_0} \tag{2.13}$$
$$\text{subject to} \quad a_{i1}x_1 + a_{i2}x_2 \le b_i, \; i = 1, 2, ..., m,$$
$$x_1, x_2 \ge 0.$$

Figure (2.2) shows the feasible region S of the problem (2.13). For any arbitrary real value K, if the equation $Q(x) = K$ or

$$(p_1 - Kd_1)x_1 + (p_2 - Kd_2)x_2 + (p_0 - Kd_0) = 0, \tag{2.14}$$

intersects the set of feasible solutions S, these intersection points represent the feasible solutions corresponding to the objective function value K.

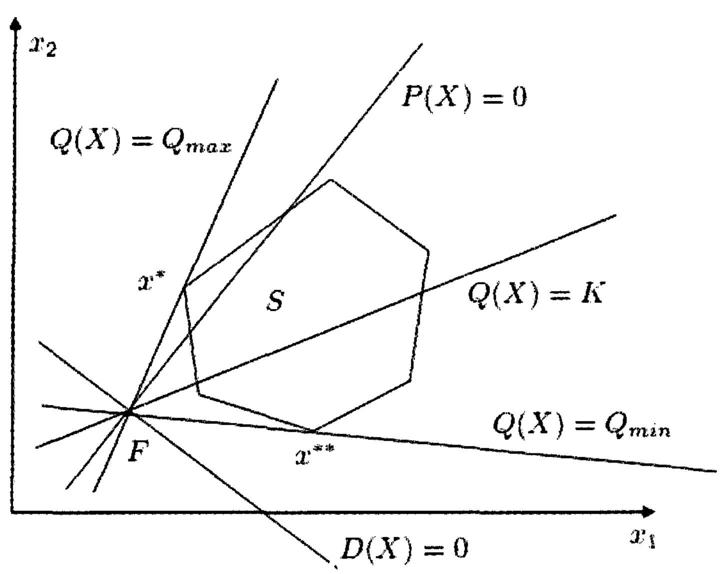

Figure 2.2 The graphical representation of an LFP problem (2.13)

For any value of the parameter K, these so-called level-lines (2.14) cross each other at point F which is an intersection point of lines $P(X) = 0$ and $D(X) = 0$. There exists a solution to Problem (2.13) if these two lines are not parallel [3].

The maximum or minimum of the LFP problem (2.13) can be found through rotating the level-lines around point F. This solution approach can be explained as follows:

Rewriting equation (2.14) we have

$$x_2 = -\frac{p_1 - Kd_1}{p_2 - Kd_2}x_1 - \frac{p_0 - Kd_0}{p_2 - Kd_2}, \tag{2.15}$$

where the slope of this line is $m = -\frac{p_1 - Kd_1}{p_2 - Kd_2}$. This slope depends on the value K of the objective function and is a monotonic function of K, because the sign of the term $\frac{dm}{dK}(\frac{dm}{dK} = \frac{d_1p_2 - d_2p_1}{(p_2 - Kd_2)^2})$ does not depend on the value of K. This sign is equal to the sign of the term $\{d_1p_2 - d_2p_1\}$ which has a constant value.

The latter means that based on the sign of the term $\{d_1p_2 - d_2p_1\}$, the value of the objective function decreases or increases by rotating level-line (2.14) around the focus point F in positive direction (counterclockwise).

In figure (2.2), rotating level-line around point F in positive direction increases the value of the objective function and leads to the maximal and minimal objective function value over set S on the points x^* and x^{**} respectively. Alternative solutions to the LFP problem (2.13) can also be explained like LP problems in a graphical way.

Optimality

In the case of a general single ratio problem, if $P(x)$ is a nonnegative concave and $D(x)$ is a positive convex, then $\frac{P(x)}{D(x)}$ is strictly quasi-convex. Therefore, we have a unique local maximum which is also global [34]. Several algorithms can be used to find this optimal solution. A single ratio concave-convex fractional program can be converted to a maximization problem with the help of the generalized Charnes & Cooper's transformation [3]. The other famous algorithm to solve LFP problems is the Dinkelbach algorithm [3]. In the rest of this section, we introduce these two widely used approaches. The simplex method can be used to solve these problems as well [3].

Charnes & Cooper's Transformation

Charnes & Cooper's Transformation (CCT) reformulates the LFP problem as an LP problem with a bounded set of feasible solutions via defining new set of variables. Back to the LFP problem (2.11), we define $t_j = \frac{x_j}{D(x)}, j = 1, 2, ..., n$ and $t_0 = \frac{1}{D(x)}$ where $D(x) = \sum_{j=1}^{n} d_j x_j + d_0$.

Taking into account the new set of variables, the objective function of the LFP problem (2.11) can be reformulated as following form:

$$\max_t L(t) = \sum_{j=0}^{n} p_j t_j \tag{2.16}$$

and the set of constraints extends to the following set:

$$\text{Subject to} \quad -b_i t_0 + \sum_{j=1}^{n} a_{ij} t_j \leq 0, \tag{2.17}$$

$$\sum_{j=0}^{n} d_j t_j = 1, \tag{2.18}$$

$$t_j \geq 0, \quad j = 1, 2, ..., n. \tag{2.19}$$

The constraint (2.18) has been added to make a connection between the original variables x_j and new variables t_j. This constraint can be provided via multiplying the term $\frac{1}{D(x)}$ to the terms of function $D(x) = \sum_{j=1}^{n} d_j x_j + d_0$.

Lemma 2.3.1 *[3] If vector $t = (t_0, t_1, ..., t_n)^\top$ is a feasible solution of problem (2.16) - (2.19), then $t_0 > 0$.*

Proof. Let us suppose the vectors

$$x' = (x'_1, x'_2, ..., x'_n)^\top, \text{and} \quad t' = (t'_1, t'_2, ..., t'_n)^\top$$

are feasible solutions to the original LFP problem (2.11) and problem (2.16), respectively. Assume that

$$t'_0 = 0, \quad \text{i.e.} \quad t' = (0, t'_1, t'_2, ..., t'_n)^\top.$$

Since vectors x' and t' are feasible solutions to their problems, (2.11) and (2.16) respectively, this follows that:

$$\sum_{j=1}^{n} a_{ij} x'_j \leq b_i, \quad i = 1, 2, ..., m, \tag{2.20}$$

$$x'_j \geq 0, \quad j = 1, 2, ..., n, \tag{2.21}$$

and

$$\sum_{j=1}^{n} a_{ij} t'_j \leq 0, \quad i = 1, 2, ..., m, \tag{2.22}$$

$$t'_j \geq 0, \quad j = 1, 2, ..., n. \tag{2.23}$$

Let us multiply each i-th constraint of system (2.22) by arbitrary positive λ and then add it to appropriate i-th constraint of the system (2.20). The same λ we will use to multiply each j-th restriction (2.23) and then to add it to the appropriate j-th constraint of (2.21), and hence we have:

$$\sum_{j=1}^{n} a_{ij}(x'_j + \lambda t'_j) \leq b_i, \quad i = 1, 2, ..., m, \tag{2.24}$$

$$(x_j^{'} + \lambda t_j^{'}) \geq 0, \quad j = 1, 2, ..., n. \tag{2.25}$$

It means that vector $x + \lambda t^{'}$ is a feasible solution of the original LFP problem for any positive λ. But λ may be as large as required, and hence it follows that feasible set S is unbounded. The latter contradicts our assumption that S is a bounded set. □

Theorem 2.3.1 *[3] If vector $t^* = (t_0^*, t_1^*, ..., t_n^*)^\top$ is an optimal solution of problem (2.16) - (2.19), then vector $x^* = (x_0^*, x_1^*, ..., x_n^*)^\top$ is an optimal solution of the original LFP problem (2.11),where*

$$x_j^* = \frac{t_j^*}{t_0^*}, \quad j = 1, 2, ..., n. \tag{2.26}$$

Proof. Since vector t^* is the optimal solution of problem (2.16), it follows that:

$$L(t^*) \geq L(t), \quad \forall t \in T, \tag{2.27}$$

where T denotes a feasible set of solution of problem (2.16). Let us suppose that vector x^* is not an optimal solution of the maximization LFP problem (2.11). Hence, there exists some another vector $x^{'} \in S$, such that $Q(x^{'}) \geq Q(x^*)$. But at the same time we have

$$\begin{aligned} Q(x^*) &= \frac{\sum_{j=1}^n p_j x_j^* + p_0}{\sum_{j=1}^n d_j x_j^* + d_0} && (2.28) \\ &= \frac{\sum_{j=1}^n p_j \frac{t_j^*}{t_0^*} + p_0}{\sum_{j=1}^n d_j \frac{t_j^*}{t_0^*} + d_0} && (2.29) \\ &= \frac{\sum_{j=1}^n p_j t_j^* + p_0 t_0^*}{\sum_{j=1}^n d_j t_j^* + d_0 t_0^*} && (2.30) \\ &= \frac{\sum_{j=1}^n p_j t_j^* + p_0 t_0^*}{1} = L(t^*). && (2.31) \end{aligned}$$

It means that

$$Q(x^{'}) \geq L(t^{*}). \tag{2.32}$$

Since vector $x^{'}$ is a feasible solution to the original LFP problem (2.11), it is easy to show that vector

$$t^{'} = (t_0^{'}, t_1^{'}, ..., t_n^{'})^{\top}, \text{where } t_0^{'} = \frac{1}{D(x^{'})}, t_j^{'} = \frac{x_j^{'}}{D(x^{'})}, \quad j = 1, 2, ..., n,$$

is a feasible solution of (2.16) and

$$L(t^{'}) \geq L(t^{*}).$$

But the latter contradicts our assumption that vector t^{*} is an optimal solution of the maximization problem (2.16). It means that vector x^{*} is an optimal solution of the maximization LFP problem (2.11). □

We illustrate by a numerical example the above-mentioned considerations. Consider the following single ratio problem.

Example:

$$\begin{aligned} \max_{x} \quad & Q(x) = \frac{x_1 + 3x_2 + 2.5x_3 + 6}{2x_1 + 3x_2 + 2x_3 + 12} \\ \text{subject to} \quad & x_1 + 2x_2 + 2.5x_3 \leq 40, \\ & 2x_1 + 2x_2 + 2x_3 \leq 60, \\ & x_j \geq 0, j = 1, 2, 3. \end{aligned}$$

The results for this given problem are as follows:

$$x^{*} = (0, 0, 16)^{\top}, \quad P(x^{*}) = 46, \quad D(x^{*}) = 44, \quad Q(x^{*}) = \frac{23}{22}.$$

The transformation of the above example based on the Charnes & Cooper's method can be denoted as:

$$\begin{array}{ll} \max\limits_{t} & L(x) = 6t_0 + 1t_1 + 3t_2 + 2.5t_3, \\ \text{subject to} & 12t_0 + 2t_1 + 3t_2 + 2t_3 = 1, \\ & -40t_0 + 1t_1 + 2t_2 + 2.5t_3 \leq 0, \\ & -60t_0 + 2t_1 + 2t_2 + 2t_3 \leq 0, \\ & t_j \geq 0, j = 1, 2, 3. \end{array}$$

The solution of this LP problem is:

$$t^* = (\frac{1}{44}, \frac{0}{44}, \frac{0}{44}, \frac{16}{44}), \quad L(t^*) = \frac{23}{22},$$

which has the same optimal value as that of the original problem.

Dinkelbach's algorithm

This algorithm is a parametric approach to solve the LFP problems. This algorithm reduces the solution of the LFP problem to the solution of a sequence of LP problems. The theoretical foundation of this algorithm is based on the following theory. Consider the following formulation:

$$\begin{array}{ll} \max\limits_{x \in S} & Q(x) = \dfrac{P(x)}{D(x)} = \dfrac{\sum_{j=1}^{n} p_j x_j + p_0}{\sum_{j=1}^{n} d_j x_j + d_0} \\ & x_j \geq 0, \quad j = 1, 2, \ldots, n, \end{array} \tag{2.33}$$

where S is a nonempty compact set of R^n and $P(x)$ and $D(x)$ are linear functions over S, and $D(x) > 0, \forall x \in S$.

Theorem 2.3.2 *[3] Vector x^* is an optimal solution of the LFP problem (2.33) if and only if*

$$F(\rho^*) = \max\{P(x) - \rho^* D(x)\} = 0, \forall x \in S, \tag{2.34}$$

where

$$\rho^* = \frac{P(x^*)}{D(x^*)}. \tag{2.35}$$

Proof. If vector x^* is an optimal solution of problem (2.34) then

$$P(x) - \rho^* D(x) \leq P(x^*) - \rho^* D(x^*) = 0,\ \forall x \in S. \tag{2.36}$$

This means that vector x^* is an optimal solution of LFP problem (2.33).

Conversely, if vector x^* is an optimal solution of problem (2.33) then

$$\rho^* = \frac{P(x^*)}{D(x^*)} \geq \frac{P(x)}{D(x)},\ \forall x \in S. \tag{2.37}$$

The latter means that

$$P(x) - \rho^* D(x) \leq 0,\ \forall x \in S. \tag{2.38}$$

Taking into account equality (2.35) we obtain

$$\max_{x \in S}\{P(x) - \rho^* D(x)\} = 0. \tag{2.39}$$

□

The Dinkelbach algorithm's steps are:

- Step 1: Take $x^{(0)} \in S$, compute $\rho^{(1)} := \frac{P(x^{(0)})}{D(x^{(0)})}$ and set $k := 1$.
- Step 2: Determine $x^{(k)} := \text{argmax}_{x \in S}\{P(x) - \rho^{(k)} D(x)\}$.
- Step 3: If $F(\rho^{(k)}) = 0$ then $x^* = x^k$ is an optimal solution; stop.
- Step 4: Set $\rho^{(k+1)} := \frac{P(x^{(k)})}{D(x^{(k)})}$; set $k := k + 1$; goto step 2.

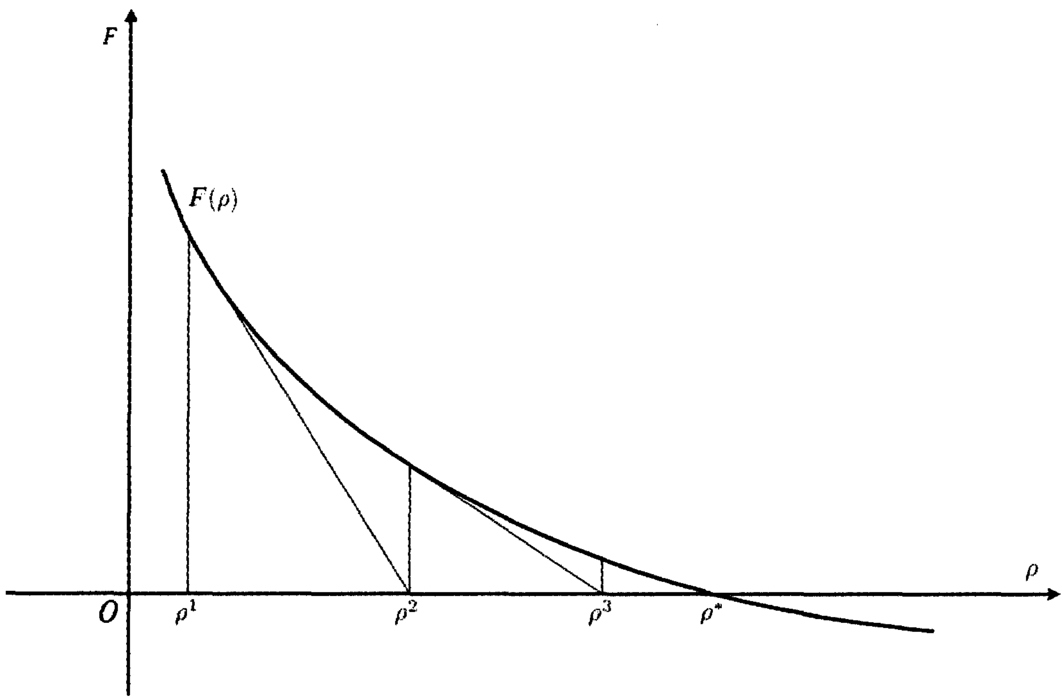

Figure 2.3 Illustration of the sequence $\{\rho^k\}$ generated by the Dinkelbach algorithm

As we have supposed that $D(x) > 0, \forall x \in S$, then $\frac{\partial F(\rho)}{\partial \rho} = -D(x) < 0$, which means that $F(\rho)$ is strictly decreasing in ρ. This algorithm's convergence rate to an optimal solution is at least linear [13]. Figure (2.3) illustrates the sequence $\{\rho^k\}$ generated by the Dinkelbach algorithm. To make it clearer, we implement the Dinkelbach algorithm on the following example.

Example:

$$\begin{aligned} \max_x \quad & Q(x) = \frac{2x_1 + x_2 + 8}{4x_1 + x_2 + 16} \\ \text{subject to} \quad & 2x_1 + 3x_2 \leq 12, \\ & 2x_1 + 4x_2 \leq 16, \\ & x_j \geq 0, j = 1, 2. \end{aligned}$$

Step 1: $x = (0,0)^\top$ satisfies all constraints of the problem, so $x^{(0)} = (0,0)^\top$ and

$$\rho^{(1)} := \frac{P(x^{(0)})}{D(x^{(0)})} = \frac{1}{2}.$$

Step 2: By solving the following LP problem

$$\max_x \{P(x) - \rho^{(1)} D(x)\} = P(x) - \frac{1}{2} D(x) = \frac{1}{2} x_2,$$

subject to the original problem's constraints, we obtain:

$$x^{(1)} = (0,4)^\top.$$

Step 3: Since, $F(\rho^{(1)} = \frac{1}{2}) = 2, (\neq 0)$ we continue the algorithm.

Step 4: $\rho^{(2)} := \frac{P(x^{(1)})}{D(x^{(1)})} = \frac{3}{5}, k = 2.$

Then , the maximum of $\{P(x) - \rho^{(2)}D(x)\} = P(x) - \frac{3}{5}D(x) = \frac{-2}{5}x_1 + \frac{2}{5}x_2 - \frac{8}{5}$, subject to the original problem's constraint, is $x^{(2)} = (0,4)^\top$. Since $F(\rho^{(2)} = \frac{3}{5}) = 0$, then $x^* = x^2$ is the optimal solution with the optimal objective function value $Q(x^*) = \frac{3}{5}$

Dinkelbach-type 2 algorithm

An improvement of the Dinkelbach algorithm by attempting to make the parametric function convex in a neighborhood of the optimal value has been proposed by Crouzeix et al. [14]. Their reformulation was based on the following theorem:

Theorem 2.3.3 *[14] Vector x^* is an optimal solution of the LFP problem if and only if*

$$F(\rho^*) = \max\{\frac{P(x) - \rho^* D(x)}{D(x^*)}\} = 0, \forall x \in S, \tag{2.40}$$

where

$$\rho^* = \frac{P(x^*)}{D(x^*)}. \tag{2.41}$$

As we do not know $D(x^*)$ so far, we use $D(x^{k-1})$ in $k-th$ iteration.

This algorithm's steps are:

- Step 1: Take $x^{(0)} \in S$, compute $\rho^{(1)} := \frac{P(x^{(0)})}{D(x^{(0)})}$ and set $k := 1$.
- Step 2: Determine $x^{(k)} :=$ argmax$\{\frac{P(x)-\rho^k D(x)}{D(x^{k-1})}\}$.
- Step 3: If $F(\rho^{(k)}) = 0$ then $x^* = x^k$ is an optimal solution; stop.
- Step 4: Set $\rho^{(k+1)} := \frac{P(x^{(k)})}{D(x^{(k)})}$; set $k := k+1$; goto step 2.

This algorithm converges to the optimal solution super-linearly [13].

Performance of Dinkelbach algorithm on 0-1 fractional programming

Matsui et al. [29] show that the Dinkelbach algorithm , in the worst case will solve an LFP problem with binary variables by using a maximum of $\mathbf{O}(\log(nM))$ iterations where

$$M = \max\{\max_{i=1,2,\ldots,n} |p_i|, \max_{i=1,2,\ldots,n} |d_i|, 1\}.$$

The simplex method for solving LFP problems

In 1960, Bela Martons [27] extended the simplex method to solve the LFP problems. According to the simplex method in LP, the LFP is solvable if the feasible set is not empty and the objective function has a finite upper bound over set S. Further information about this method is available in Bajalinove [3].

2.3.2 Linear Fractional Programming - Sum of Several Ratios

In the formulation (2.10), if $l \geq 2$, $S \subseteq R^n$ is nonempty and $D(x)$ is positive for all $x \in S$, we are facing a maximization of sum of ratios problem. This kind of formulation has numerous important applications in practical real world problems. These problems appear when at the same time we want to optimize a weighted sum of several rates. The numerator and denominator of these ratios present different elements such as cost, profit, input, output, capital, expense, time, and so on.

Applications

Almogy and Levin [2] formulate a multistage stochastic shipping problem to be a sum of ratios problem. The next application of this problem was in a clustering problem. Rao [31] formulates the clustering problem, one of the most common mathematical programming problems, by a sum of ratios. This problem arises when we want to partition a given set of entities into a number of mutually exclusive and exhaustive clusters. Here the objective is to find a minimum sum of an average squared distance within groups. Schaible and Simchi-Levi [16] present the minimization of the mean response time in queueing location problems as the minimization of a sum of ratios as well.

There are some other areas in which this problem arises, such as material control problems [33], production lot sizing with material handling cost consideration [21], bond portfolio optimization problems [23], hospital management [28], and many other applications have been studied in Chen et al. [11].

Theoretical aspect

Unfortunately, there are not significant properties for mathematical aspects of the sum of ratios. Unlike a single ratio, the property of being quasi-concave is not valid any more in the sum of ratios. Therefore, a local optimum is not generally global, even when all of our ratios and functions are linear. Hence, this problem should be considered in the context of global optimization *i.e.* there are multiple local optimum points which are not globally optimum.

Craven [10] shows that when we have just two linear ratios, the optimum point is often on the vertex or an edge of the convex feasible region. He shows that generally for more than two ratios, the optimum point is on the boundary of the convex feasible region, if it exists, and finally, the Duality Theorem for the sum of several ratios was proposed first by Scott and Jefferson [35].

Complexity

Freund and Jarre [19] show that even in the case of concave-convex ratios and concave functions, the sum of the ratios problem is NP-hard.

Algorithms

As expected, since we do not have strong mathematical structure for the sum of ratios problem, efficient algorithmic approaches for this problem are limited. Craven [10] propose a simplex algorithm for just having two ratios. However, when more than two ratios are present, there are several other algorithms. For the problem with a few ratios, such as exactly three ratios, and some other special cases, Konno et al. [22] [24] propose some parametric and heuristic approaches to be employed to overcome this complexity. Falk and Palocsay [18] propose a new method by using the image space analyzing concept. They make the problem simpler by assigning each of the ratios to a new variable defined in the image space with certain directions. Afterwards, they find a global optimum by optimizing in this direction.

Konno and Fukaishi [22] assign new variables for each one of the ratios. Subsequently, they transform the nonlinearity from the objective function to the multiplicative constraints and they apply a branch and bound algorithm to solve the problem. Benson [8] extends the method proposed by Konno and Fukaishi [22] to solve a sum of ratios problem with nonlinear terms in the numerator and denominator.

In the case of just two variables and more ratios, Chen et al. [11] propose an efficient algorithm by using computational geometry. Kuno [25] propose a branch and bound algorithm to solve the sum of several ratios problem. To perform a bounding operation, they associate for the numerator and denominator of a new variable and they do the bounding operation in a defined $2D$-dimensional space.

Freund and Jarre [19] propose a procedure to convert the minimization problem of sum of l ratios to the minimization problem of a function with l variables, where the function values are given by the solution of specific convex subproblems. Afterwards they propose an interior point method to find the global minimum for convex programs.

More recently, Benson [7] presents a branch and bound algorithm to globally solve the sum of fractional ratios where they are transformed to an equivalent concave minimization problem. The main advantage of this algorithm is that at every iteration, the subproblems have the same size in the number of variables and the subproblems are different in only the coefficients, so an optimal solution for one subproblem can be a good, feasible solution for the next subproblem.

Most recently, Wu et al. [42], for the case in which the number of ratios is small, propose an efficient method based on the transformation of the objective function to the image space. In this method, they reduce the problem to the sequence of single ratio problems and then apply a stochastic search algorithm to the transformed image space to find the solution for the reduced problems. This algorithm is based on the Electromagnetism-like Mechanism (EM) method.

2.4 Conclusion

In this chapter, we presented a summary of choice-based revenue management problems followed by some theoretical aspects of linear fractional programming problems in the forms of single ratio or sum of several ratios. Several algorithms and numerical results were presented.

CHAPTER 3

PROBLEM DESCRIPTION AND SOLUTION APPROACHES

3.1 Introduction

According to the previous chapters, one of our main challenges in choice-based revenue management modeling is how to construct a relevant choice model with our given data. In this chapter, we are considering the more general form of the choice-based, deterministic, linear programming model proposed by Gallego et al. [20] for overlapping segments. As we develop a column generation algorithm to solve it on a real-sized airline or railroad network, we face a linear fractional programming subproblem which is NP-Hard. We study available solution approaches and we provide a heuristic with high quality results to tackle this complexity.

3.2 Problem Description

In this research, we use the terminology of the airline application as representative of the problem. We continue by providing some definitions which are used in the upcoming parts of this chapter.

- **Market:** An origin-destination pair, between which passengers wish to travel.
- **Itinerary:** A specific sequence of legs on which passengers travel from their origin to their ultimate destination.
- **Fare Classes:** Different prices for the same travel service, usually distinguished from one another by the set of restrictions imposed by the firms.

- **Product:** Generally defined by an itinerary and fare-class combination.
- **Consideration set:** A subset of products provided by the firm that a customer views as an option.
- **Segment:** Customers, based on their preferences, are divided to different segments which each segment is defined by a consideration set of products.

Starting to build an appropriate model, our given data include:

- Available network for transportation with its properties.
- Possible products with their properties and fares.
- Customers' segmentations based on their preferences.
- Customers' behavior estimation parameters.
- Booking horizon.

The objective should be to find the set of alternative products which firms should decide to offer to customers at the time they are going to make a decision. The prices are fixed and the firm wants to maximize its revenue.

As mentioned before, our work is mostly motivated by the work of Vulcano et al. [9]. This model is a developed version of works done by Gallego et al. [20] and Van Ryzin and Liu [40]. Gallego et al. [20] propose a new model for the customer choice-based deterministic linear programming problem. In their model, they suppose the firm has the flexibility to offer different products for the same market to satisfy their demand. One of the main limitations in their formulation is that they did not define any segmentation for customers.

In the van Ryzin and Liu's model [40], they suppose that each customer belongs to one segment, where the segments are defined by a separated consideration set of products. This assumption causes a consequential improvement in the firm's revenue. Finally, Vulcano et al. [9] consider a more general case of this model for overlapping segments in which one product can belong to two different segments at the same time. Certainly, the preferences of customers for this product is different in the corresponding segments (*i.e.* one person with an economic class preference can finally choose a high class ticket, but with different and less preference). It is clear that by considering overlapping segments, we have a better modeling and understanding of customer behavior, which will increase the firm's revenue.

Example 3.2.1

Let us consider a very small airline network with three cities, *e.g.* Montreal, Toronto and Vancouver to make the problem clearer. The firm is offering two fare classes, low and high, for each flight (leg). Figure (3.1) illustrates this network.

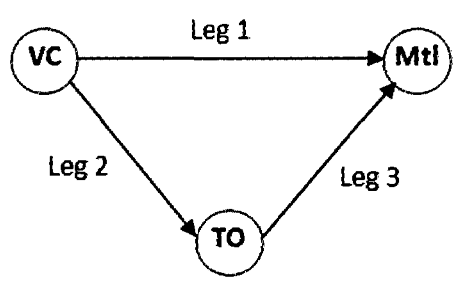

Figure 3.1 Simple airline network illustration

The legs have the capacities $c = (10, 5, 5)$ respectively. According to the following Table (3.1), we have definitions of available products defined by an itinerary and fare-class combination.

Table 3.1 Product definition for the small network example

Product	O-D	Class	Fare
1	VC $\rightarrow$ MTL	High	1500
2	VC $\rightarrow$ TO $\rightarrow$ MTL	High	900
3	VC $\rightarrow$ TO	High	600
4	TO $\rightarrow$ MTL	High	600
5	VC $\rightarrow$ MTL	Low	900
6	VC $\rightarrow$ TO $\rightarrow$ MTL	Low	600
7	VC $\rightarrow$ TO	Low	400
8	TO $\rightarrow$ MTL	Low	400

Based on customers' preferences for price and time, we divide them into five different segments. This definition is shown in Table (3.2). As we can see in the description column, price and time are the main factors to categorize customers. The second column shows an arrival probability of the corresponding segment. Columns three and four correspond to the consideration set and preference values for relevant products including no-purchase preference in the last coordinate, respectively.

Table 3.2 Segment definition for the small network example

Segments	λ_l	Cns. Set	Prf. vector	Description
1	0.10	$\{1,2\}$	(9,5,4)	Pr. insensitive (VC $\rightarrow$ Mtl)
2	0.20	$\{5,6\}$	(9,6,2)	Pr. sensitive (VC $\rightarrow$ Mtl)
3	0.30	$\{3,7\}$	(4,7,3)	Pr. sensitive (VC $\rightarrow$ TO)
4	0.25	$\{4,8\}$	(5,8,3)	Pr. sensitive (TO $\rightarrow$ Mtl)
5	0.15	$\{1,5\}$	(6,9,3)	Pr. sensitive, Non stop (VC $\rightarrow$ Mtl)

Price insensitive customers belong to segment 1. This would be business travelers who prefer to travel directly from Vancouver to Montreal but may accept a stop in Toronto, while leisure travelers who want to travel non-stop directly from Vancouver to Montreal belong to segment 5. As you can see in Table (3.2), product 1 is common to both segments because it takes into consideration overlapping segments in customer behavior. Segment 2 consists of leisure travelers who want to travel from Vancouver to Montreal, who consider either non-stop and correspondent products. Segments 3 and 4 describe price-sensitive customers that want to travel from Vancouver to Toronto or Toronto to Montreal, respectively.

The firm wants to maximize its expected revenue by having a policy to offer a set of products S at any time t during a booking horizon of $T = 30$ periods.

3.3 Choice-based deterministic linear programming model

Notations

To define our model, consider a network with m resources (legs) providing n products. $N = \{1, 2, .., n\}$ denotes the set of products and r_j is the associated revenue (fare) for product $j \in N$. We study capacity usage by defining vector $c = (c_1, c_2, ..., c_m)$ which denotes the initial capacities of resources (legs). Resource use according to the corresponding products is presented by defining an incidence matrix $A = [a_{ij}] \in B^{m \times n}$.

The matrix entries are defined by:

$$a_{ij} = \begin{cases} 1, & \text{if resource } i \text{ is used by product } j, \\ 0, & \text{otherwise.} \end{cases}$$

A_j, the j-th column of A, denotes the incidence vector for product j and notation $i \in A_j$ indicates that product j is using resource i (that is, $a_{ij} = 1$). Note that one product can use more than one resource. Time has discrete periods and runs forward until a finite number T, $t = 1, 2, ..., T$ and it is undertaken that we have at most one arrival for each period of time and each customer can buy only a single product. The length of time is a fundamental decision. If we take a unit of time that is very small such as a few seconds, then in a given O-D market, in just a few periods we will have bookings and most of them will be no-purchase. Vulcano and Van Ryzin [41], based on numerical experiments on airline networks, suggest dividing the day to $T = 140$ small time periods (considering approximately every 10 minute as a period of time) and we use λ to denote the probability of having an arrival in a period of time.

We divide customers into L different segments. A consideration set $C_l \subseteq N, l = 1, 2, ..., L$ is used to describe each segment. Here we can make the difference of this model clearer with previous works on customer choice-based modeling. Gallego et al. [20] considers a unique segment $C_1 = N$ and unlike Van Ryzin and Liu [40] we can have overlapping segments, that is, $C_l \bigcap C_{l'} \neq \emptyset$ for certain $l \neq l'$.

If we have one arrival, p_l represents the probability that an arriving customer belongs to segment l with $\sum_{l=1}^{L} p_l = 1$. We consider a Poisson process of arriving streams of customers from segment l with rate $\lambda_l = \lambda p_l$ and total arriving rate of $\lambda = \sum_{l=1}^{L} \lambda_l$.

In each period of time t, the firm should decide about his offer set (*i.e.* a subset $S \subset N$ of products that the firm makes available for customers). If set S is offered, the deterministic quantity $P_j(S)$ indicates the probability of choosing product $j \in S$ and $P_j(S) = 0$ if $j \notin S$. By total probability law, we have $\sum_{j \in S} P_j(S) + P_0(S) = 1$, where $P_0(S)$ indicates the no-purchase probability.

As already stated, we use a multinomial logit (MNL) model to find customer choice probabilities. According to a MNL choice model, vector $v_l \geq 0$ is a customer's preference vector for available products in consideration set C_l and v_{l0} represents the no-purchase preference. We let $P_{lj}(S)$ denote the probability of selling product $j \in C_l \bigcap S$ to a customer from segment l when set S is offered. So, customer choice probability can be expressed as follows:

$$P_{lj}(S) = \frac{v_{lj}}{\sum_{h \in C_l \bigcap S} v_{lh} + v_{l0}}. \tag{3.1}$$

It can be obtained from equation (3.1) that $P_{lj}(S) = 0$ if $v_{lj} = 0$ which can be a result of $j \notin C_l$ or $j \notin C_l \bigcap S$. We assume $v_{l0} > 0$ for all segment $l = 1, 2, ..., L$. In the more general case, as a firm cannot recognize the corresponding segment of an arrival in advanced, we consider $P_j(S)$, the probability that the firm sells product j to an arriving customer as:

$$P_j(S) = \sum_{l=1}^{L} p_l P_{lj}(S). \tag{3.2}$$

The expected revenue, by offering set $S \subset N$ from an arriving customer is given by:

$$R(S) = \sum_{j \in S} r_j P_j(S). \tag{3.3}$$

Given that we offer set S, let $P(S) = (P_1(S), ..., P_n(S))^\top$ be the vector of purchase probabilities and A the incidence matrix of resource use by products. Then the vector of capacity consumption probabilities $Q(S)$ is given by:

$$Q(S) = AP(S), \tag{3.4}$$

where $Q(S) = (Q_1(S), ..., Q_m(S))^\top$ and $Q_i(S)$ indicates the probability of using a unit of capacity on leg $i, i = 1, 2, ..., m$.

Linear programming formulation

The firm's decision consists of deciding that at any period of time t, which set of products should be offered, while it could not distinguish each customer's related segment in advance. However, as choice probabilities are time-homogeneous and demand is deterministic, it only matters how many times each set S is offered and knowing during exactly which period is not important and the variable $t(S)$ represents the number of periods during which set S is going to be offered. Another assumption is that we let variable $t(S)$ to be continuous as well (*i.e.* the firm could offer a set S for a whole or a fraction of a period of time). The model's objective is to maximize the firm's revenue by deciding the number of periods of time for each set of products. Corresponding to formulation (3) in Vulcano et al. [9] this leads to the following LP formulation:

$$\begin{aligned} V^{CDLP} = \quad & \max \sum_{S \subset N} \lambda R(S) t(S) \\ \text{subject to} \quad & \sum_{S \subset N} \lambda Q(S) t(S) \leq c, \\ & \sum_{S \subset N} t(S) \leq T, \\ & t(S) \geq 0, \forall S \subset N. \end{aligned} \tag{3.5}$$

There are $m + 1$ constraints in the formulation (3.5), where the first m constraints are related to availability of capacity and the last one is for time availability. Because of the number of constraints ($m + 1$), we could have a maximum of $m + 1$ variables with a positive value in the base.

Van Ryzin and Liu [40] prove that since demand and capacity are scaled up proportionately, the revenue obtained under the CDLP model is asymptotically optimal for the original stochastic network choice model.

Let now return to example (3.2.1). We could have $2^8 - 1 = 255$ possible non-empty sets of combination of 8 available products. By solving the CDLP model (3.5) for our example, the optimal sets to offer will be $S_1 = \{1, 3, 7\}$, with $t(S_1) = 10$ periods, $S_2 = \{1, 2, 3, 7\}$, with $t(S_2) = 4$ periods, and $S_3 = \{1, 3, 4, 6, 7\}$, with $t(S_3) = 16$ periods. As we can see in the optimal solution, products 1, 3 and 7, which are the products with the highest revenue and demand, are all offered in the whole booking horizon. Products 4 and 6 are offered during 16 periods and product 2 is offered only 4 periods of time and products 5 and 8 are never offered. There are some remarks which should be mentioned here about CDLP model and its optimal solution.

First, we should decide how to apply the solution of the CDLP model in our real problem and assign a start and end time to offer each product. As mentioned before, the CDLP model's solution does not give us a sequence of products and times. However, to order the offer sets, various heuristic approaches can help us. Van Ryzin and Liu [40] developed an efficient decomposition heuristic to overcome this problem.

Second, in the formulation (3.5) there are an exponential number of primal variables. This means that a problem with n products, has $2^n - 1$ possible non-empty subsets of products of set N. In spite of an enormous number of variables for practical real world problems which makes it impossible to enumerate all offer sets, there are at most $m + 1$ constraints. This leads to the idea of using a column generation technique to solve real world practical problems.

3.4 Using column generation to solve the CDLP model

Gallego et al. [20] suggest using column generation to solve real world CDLP models. This algorithm's steps are described in the following steps and in Figure (3.2) as well:

- Step 1: Start by solving a reduced LP (RPP); that is, just considering a limited number of columns (subsets) instead of enumerating all of them.
- Step 2: Construct a subproblem by using the dual solution of RPP to find a column with the most positive reduced cost.
- Step 3: Add the column with a positive reduced cost to RPP and solve it again.
- Step 4: If there is no column with a positive reduced cost, then the current solution is optimal.

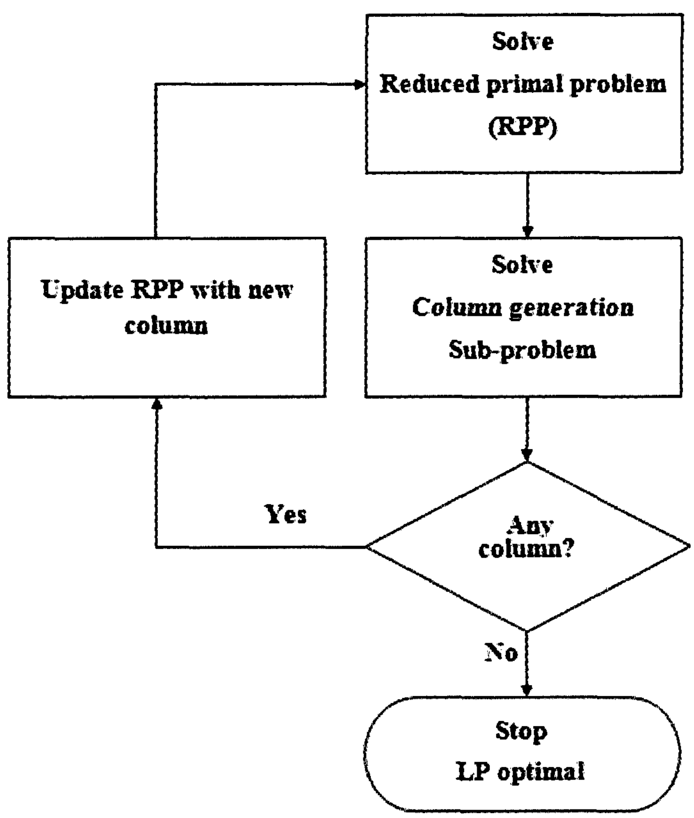

Figure 3.2 Column generation steps to solve the CDLP model.

Let return to the original CDLP model (3.5), but just with limited initial columns (subsets) indicated by $\mathcal{N} = \{S_1, S_2, ..., S_k\}$. This takes us to the reduced CDLP model as follows:

$$\begin{aligned} V^{CDLP-R} \quad &= \quad \max \sum_{S \in \mathcal{N}} \lambda R(S) t(S) && (3.6) \\ \text{subject to} \quad & \sum_{S \in \mathcal{N}} \lambda Q(S) t(S) \leq c, \qquad (\pi) \\ & \sum_{S \in \mathcal{N}} t(S) \leq T, \qquad (\sigma) \\ & t(S) \geq 0, \forall S \in \mathcal{N}. \end{aligned}$$

Let $\pi \in \mathbf{R}^m$ be to the dual prices for the first m-dimensional capacity constraints and $\sigma \in \mathbf{R}$ the dual price for the unidimensional time constraint. Now for the next step in the column generation algorithm, we construct a column generation subproblem to find the next column with the most positive reduced cost to add to our set collection $\mathcal{N}$ which is not included yet. This column is obtained by solving the following subproblem:

$$\max_{S \subseteq N} \{\lambda R(S) - \lambda \pi^\top Q(S) - \sigma\} = \max_{S \subseteq N} \{\lambda R(S) - \lambda \pi^\top Q(S)\} - \sigma. \qquad (3.7)$$

Afterwards, to explicit the formulation (3.7), a binary vector $y \in \mathbf{B}^n$ is defined as follows. Suppose a set S is offered now, then we denote:

$$y_j = \begin{cases} 1, & \text{if } j \in S, \\ 0, & \text{otherwise.} \end{cases}$$

After introducing the binary variables y_j, the problem (3.7) can be expressed as follows:

$$\max_{y\in\{0,1\}^n}\left\{\sum_{l=1}^{L}\lambda_l\frac{\sum_{j\in C_l}(r_j - A_j^\top\pi)v_{lj}y_j}{\sum_{i\in C_l}v_{li}y_i + v_{l0}}\right\} - \sigma, \tag{3.8}$$

or equivalently,

$$\max_{y\in\{0,1\}^n}\left\{\sum_{j=1}^{n}\left(r_j - A_j^\top\pi\right)y_j\left(\sum_{l=1}^{L}\frac{\lambda_l v_{lj}}{\sum_{i\in C_l}v_{li}y_i + v_{l0}}\right)\right\} - \sigma. \tag{3.9}$$

Note that we assume $v_l \geq 0$ and $v_{l0} > 0, \forall l$, to be certain that our denominator is greater than zero all the time. If the problem (3.9) has a positive optimal value, then the optimal solution for the problem (3.9) will be the next entering column (subset) to the reduced primal problem (3.6). Then we update the reduced CDLP (3.6) with the new column and iterations are continued. Finally, if there was no solution for the problem (3.9) with a positive value, then the current solution for the reduced CDLP problem (3.5) is optimal.

Complexity of the column generation subproblem

The 0-1 Fractional Programming Problem (3.9) can be considered as a special case of the sum of ratios problem with more firmly connected variables. Vulcano et al. [9] proved that the minimum vertex problem, which is known to be NP-Hard, can be reduced to the Problem (3.9); hence Problem (3.9) is an NP-hard problem [Theorem 1,[9]].

3.5 Solution approaches for the column generation subproblem

In this section, we study different solution approaches for the column generation subproblem starting by an exact method, followed by a greedy heuristic proposed by Vulcano et al. [9]. Finally, we propose a new efficient heuristic to face the complexity of the subproblem.

Exact method - Mixed integer programming (MIP) reformulation

One way to solve the column generation subproblem is to reformulate it as a MIP problem [9]. Consider the formulation (3.9). We start by defining new variables x_l, $l = 1, ..., L$ as follows:

$$x_l = \frac{1}{\sum_{i \in C_l} v_{li} y_i + v_{l0}}. \tag{3.10}$$

Afterwards, substituting x_l in formulation (3.9) leads to the following problem:

$$\max \sum_{l=1}^{L} \sum_{j \in C_l} \lambda_l (r_j - A_j^\top \pi) v_{lj} y_j x_l \tag{3.11}$$

$$\text{subject to} \quad x_l v_{l0} + \sum_{i \in C_l} v_{li} y_i x_l = 1, \quad l = 1, ..., L, \tag{3.12}$$

$$y_j \in \{0, 1\}, \quad j \in N, \tag{3.13}$$

$$x_l \geq 0, \quad l = 1, ..., L. \tag{3.14}$$

It can be seen that there are nonlinear terms $y_i x_l$ appearing in (3.11) and (3.12). These terms can be linearized by using the theorem proposed by Wu [43]: a polynomial mixed 0-1 term $z = xy$, where x is a continuous variable and y is a binary variable, can be represented by the following linear system:

$$x - z \leq K - Ky, \tag{3.15}$$

$$z \leq Ky, \tag{3.16}$$

$$z \leq x, \tag{3.17}$$

$$z \geq 0. \tag{3.18}$$

Let $z_{li} = x_l y_i$. Replacing new variables, problem (3.9) can be rewritten as follows:

$$\begin{aligned} \max \sum_{l=1}^{L} \sum_{j \in C_l} \lambda_l (r_j - A_j^\top \pi) v_{lj} z_{lj} & & & \\ \text{sujet à} \quad x_l v_{l0} + \sum_{i \in C_l} v_{li} z_{li} &= 1, & & \forall l, l = 1, ..., L \\ x_l - z_{li} &\leq K - K y_i, & & \forall l, l = 1, ..., L, i \in C_l, \\ z_{li} &\leq x_l, & & \forall l, l = 1, ..., L, \; i \in C_l, \\ z_{li} &\leq K y_i, & & \forall l, l = 1, ..., L, \; i \in C_l, \\ y_j \in \{0, 1\}, \; x_l \geq 0, & \quad z_{li} \geq 0. & & \end{aligned} \tag{3.19}$$

K should be a number greater than the maximum value that variable x can take. As we have defined $x_l = \frac{1}{\sum_{i \in C_l} v_{li} y_i + v_{l0}}$ and y_j only takes binary values, it is enough to take $K \geq \frac{1}{\underline{v}}$ where $\underline{v} = \min\{v_{li} : i = 0, 1, ..., n; l = 1, 2, ..., L\}$. Any commercial MIP solvers could be used to solve this formulation.

Approximate method - Greedy heuristic

The fact that the column generation subproblem is an NP-hard optimization problem forces us to use an alternative approach that makes it possible to implement this algorithm in practical problems. Vulcano et al. [9] propose a greedy heuristic with complexity $O(n^2 L)$ based on the heuristic proposed by Prokopyev [30] to face complexity of the exact algorithm.

This heuristic starts by an empty set S, and taking into account the maximum marginal contribution to the current solution, adds progressively new products to the current set S.

The algorithm is presented in the following steps:

- Step 1: For all products j such that $r_j - A_j^\top \pi \leq 0$, set $y_j = 0$.
- Step 2: Let $S' \subset N$ be the set of products j with no assigned value for y_j.
- Step 3: Compute $j_1^* = \text{argmax}_{j \in S'} \left\{ \sum_{l=1}^{L} \frac{(r_j - A_j^\top \pi) v_{lj}}{v_{lj} + v_{l0}} \right\}$. Set $S := \{j_1^*\}, S' := S' - \{j_1^*\}$.
- Step 4: **Repeat**
  - Compute $j^* := \text{argmax}_{j \in S'} \left\{ \sum_{l=1}^{L} \lambda_l \frac{\sum_{i \in C_l \bigcap (S \bigcup \{j\})} (r_i - A_i^\top \pi) v_{li}}{\sum_{i \in C_l \bigcap (S \bigcup \{j\})} v_{li} + v_{l0}} \right\}$.
  - If Value($S \bigcup \{j^*\}$) > Value(S), then $S := S \bigcup \{j^*\}$, and $S' := S' - \{j^*\}$.

  until S is not modified.
- Step 5: For all $j \in S$, set $y_j = 1$. For $j \notin S$, set $y_j = 0$.

Approximate method - Dinkelbach-Based heuristic

As mentioned, the complexity of the column generation subproblem makes it impossible to implement exact methods to solve practical real world problems. Time and quality of the solution can be considered as two main key concepts to show the efficiency of one algorithm. Here, in this section we present an efficient heuristic method to solve the column generation subproblem.

First, we start by reducing unnecessary variables to make the problem easier. As we saw in the formulation (3.9), there are coefficients that have the same sign in different ratios and this makes variables more closely linked together. So removing unnecessary variables could help us to improve our efficiency. To do so, we use the following proposition:

Proposition 3.5.1 *In the maximization of a sum of several ratios problem without constraints, if there is a variable with a negative coefficient in all of the ratios, this variable can be removed from the problem.*

Proof. Consider the following sum of ratios problem:

$$\max_x \sum_{l=1}^{L} Q_l(x) = \sum_{l=1}^{L} \frac{P_l(x)}{D_l(x)} = \sum_{l=1}^{L} \frac{\sum_{j=1}^{m} p_{lj}x_j}{\sum_{j=1}^{m} d_{lj}x_j}, \tag{3.20}$$

where $D_l(x) > 0, \forall l = 1, ..., L$. Let suppose that for a certain $j = k$ we have $p_{lk} < 0, \forall l = 1, 2, ..., L$ and other coefficients are positive. Now we can rewrite (3.20) in the following terms :

$$\max_x \sum_{l=1}^{L} Q_l(x) = \sum_{l=1}^{L} \frac{P_l(x)}{D_l(x)} = \sum_{l=1}^{L} \frac{\sum_{j=1}^{m} p_{lj}x_j}{\sum_{j=1}^{m} d_{lj}x_j} \tag{3.21}$$

$$= \sum_{l=1}^{L} \frac{\sum_{j=1,j\neq k}^{m} p_{lj}x_j}{\sum_{j=1}^{m} d_{lj}x_j} + \sum_{l=1}^{L} \frac{p_{lk}x_k}{\sum_{j=1}^{m} d_{lj}x_j}. \tag{3.22}$$

The second term of (3.22) is always negative as $p_{lk} < 0$. Next note that:

$$\sum_{l=1}^{L} \frac{\sum_{j=1,j\neq k}^{m} p_{lj}x_j}{\sum_{j=1}^{m} d_{lj}x_j} \leq \sum_{l=1}^{L} \frac{\sum_{j=1,j\neq k}^{m} p_{lj}x_j}{\sum_{j=1,j\neq k}^{m} d_{lj}x_j}. \tag{3.23}$$

Taking into account relations (3.22) and (3.23) follows that

$$\sum_{l=1}^{L} \frac{\sum_{j=1,j\neq k}^{m} p_{lj}x_j}{\sum_{j=1}^{m} d_{lj}x_j} + \sum_{l=1}^{L} \frac{p_{lk}x_k}{\sum_{j=1}^{m} d_{lj}x_j} < \sum_{l=1}^{L} \frac{\sum_{j=1,j\neq k}^{m} p_{lj}x_j}{\sum_{j=1,j\neq k}^{m} d_{lj}x_j}. \tag{3.24}$$

The latter means that the value of the objective function increases by removing positive x_k from any solution. Hence, it is not possible to have $x_k > 0$ in an optimal solution. □

Almogy and Levin [1] tried to maximize a sum of ratios problem by transforming the problem (3.20) to an equivalent parametric problem. Nonetheless, Falk and Palocsay [17] by a numerical example showed that this algorithm does not work in general. By using the idea of Almogy and Levin and Dinkelbach's algorithm, we develop a heuristic to simplify the computational steps in the column generation algorithm and finding an appropriate solution in a polynomial time.

To do so, considering problem (3.20) we define the parametric problem as follows:

$$F(\rho^*) = \max_x \left\{ \sum_{l=1}^{L} \frac{P_l(x) - \rho_l^* D_l(x)}{D_l(x^*)} \right\}, \tag{3.25}$$

which is a non-increasing convex function of ρ [32]. Likewise single ratio problem and based on lagrangian multipliers, ρ_l is defined as:

$$\rho_l = \frac{P_l(x^*)}{D_l(x^*)}. \tag{3.26}$$

As we do not know $D(x^*)$ so far, we use $D(x^{k-1})$ in $k-th$ iteration. Taking into account parametric problem $F(\rho)$, the algorithm is described in the following steps:

- Step 1: Take $x^{(0)} \in S$, compute $\rho_l^{(1)} = \frac{P_l(x^{(0)})}{D_l(x^{(0)})}$, $\forall l = 1, ..., L$; and set $k := 1$.
- Step 2: Determine $x^{(k)} = \text{argmax}_{x \in s} \left\{ \sum_{l=1}^{L} \frac{P_l(x) - \rho_l^{(k)} D_l(x)}{D_l(x^{k-1})} \right\}$.
- Step 3: If $F^{(k)}(\rho^{(k)}) = 0$ then $x^* = x^{(k)}$ is our solution, Stop.
- Step 4: Set $\rho_l^{(k+1)} = \frac{P_l(x^{(k)})}{D_l(x^{(k)})}$, $\forall l = 1, ..., L$; Set $k := k + 1$; goto step 2.

As before, k denotes the iteration number and S is set of feasible solutions. Even though there is no guarantee that an optimal solution will be found with this algorithm, the heuristic works very fast and we could find an optimal solution in most of our experiments.

3.6 Conclusion

In this chapter, we considered the more general form of the choice-based, deterministic, linear programming model proposed by Gallego et al. [20] for overlapping segments. Developing a column generation algorithm to solve this model on a real-sized network, we faced a linear fractional programming subproblem which is NP-hard. We studied available solution approaches and we provided a heuristic with high quality results to tackle this complexity.

In the next chapter, by considering two examples, we discuss the efficiency of different solution approaches mentioned in this chapter by implementing them in practical problems.

CHAPTER 4

NUMERICAL EXAMPLES AND EVALUATION OF SOLUTION APPROACHES

In this chapter, we consider two examples for a choice-based network revenue management model with overlapping segments in which customers choose their products based on an MNL choice model. Afterwards, implementing different strategies studied in the previous chapter, we report the numerical results.

Taking into account the computational results, we evaluate different solution approaches based on the quality of the obtained solution and computational time consumed for the operations and we discuss other aspects of the obtained results.

We run our algorithms on a DELL DXP061 with a 2.1 Ghz processor, 4 GB of RAM and the operating system Windows XP Professional. We use Xpress-IVE (Mosel) version 1.18 to code and execute our algorithms. Xpress-IVE (Mosel) is a fully-functional programming language specifically designed for formulating the problem, to solve it and analyze the solution [15].

4.1 A small airline network

First, we start evaluating different heuristic and exact methods with a small airline network. This example is also considered with different details by Van Ryzin and Liu [40] and Vulcano et al. [9]. Consider a network with 4 airports and 7 flight legs. The capacities of the legs are $c = (100, 150, 150, 150, 150, 80, 80)$. The firm offers two high (H) and low (L) fares on each leg. Considering local and connecting itineraries, customers can choose among 22 available products defined by itineraries and fare class combinations. The problem consists of finding a policy, which leads us to prepare a set of products at any period of time during the booking horizon to offer to the customers while the revenue of the firm should be maximized. This airline network is illustrated in Figure (4.1) and Table (4.1) describes available products in this network.

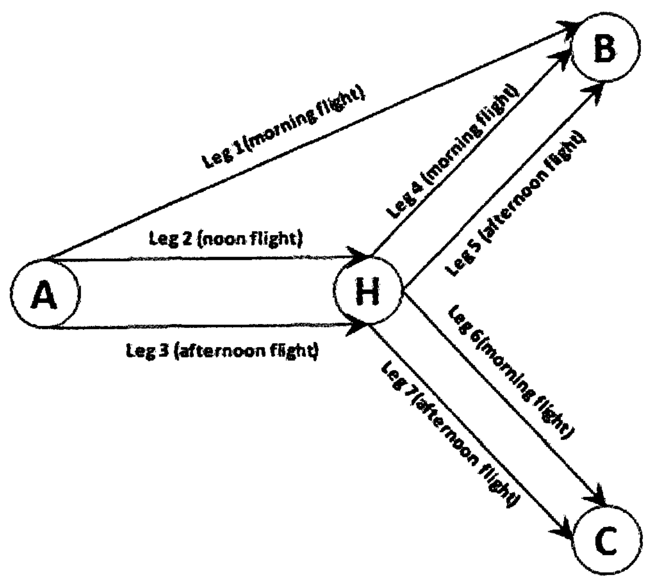

Figure 4.1 Small airline network with 2 fare classes and 7 legs

Respecting the customers' price and time sensitivities and their origin and ultimate destination, ten overlapping segments are defined in this example. These segmentations are

Table 4.1 Product description for the small network example.

Prd.	O-D	Cl.	Legs	Fare	Prd.	O-D	Cl.	Legs	Fare
1	A → B	H	1	1000	12	A → B	L	1	500
2	A → H	H	2	400	13	A → H	L	2	200
3	A → H	H	3	400	14	A → H	L	3	200
4	H → B	H	4	300	15	H → B	L	4	150
5	H → B	H	5	300	16	H → B	L	5	150
6	H → C	H	6	500	17	H → C	L	6	250
7	H → C	H	7	500	18	H → C	L	7	250
8	A → H → B	H	{2,4}	600	19	A → H → B	L	{2,4}	300
9	A → H → B	H	{3,5}	600	20	A → H → B	L	{3,5}	300
10	A →H → C	H	{2,6}	700	21	A → H → C	L	{2,6}	350
11	A → H → C	H	{3,7}	700	22	A → H → C	L	{3,7}	350

described in Table (4.2). The probability of a customer arrival for the corresponding segment is given in the second column. Columns 3 and 4 specify a corresponding consideration set and the preference values for the indicated products, respectively, and the no-purchase preference is given in the last coordinate of the preference vector. Finally, a short description of the segment is given in the last column.

Table 4.2 Segment definition for the small network example

Sgm.	λ_l	Consideration set	Pref. vector	Description
1	0.08	{1,8,9,12,19,20}	(10,8,8,6,4,4,5)	Pr. insen., early pref. A→B
2	0.2	{1,8,9,12,19,20}	(1,2,2,8,10,10,10)	Pr. sen. A→B
3	0.05	{2,3,13,14}	(10,10,5,5,5)	Pr. insen. A→H
4	0.2	{2,3,13,14}	(2,2,10,10,10)	Pr. sen. A→H
5	0.1	{4,5,15,16}	(10,10,5,5,5)	Pr. insen. H→B
6	0.15	{4,5,15,16}	(2,2,10,8,10)	Pr. sen.,slight early pref. H→B
7	0.02	{6,7,17,18}	(10,8,5,5,5)	Pr. insen., slight early pref. H→C
8	0.05	{6,7,17,18}	(2,2,10,8,10)	Pr. sens. H→C
9	0.02	{10,11,21,22}	(10,8,5,5,5)	Pr. insen., slight early pref. A→C
10	0.04	{10,11,21,22}	(2,2,10,10,10)	Pr. sen. A→C

Indeed, if the capacity of legs exceeds the corresponding demand, the problem becomes much easier to solve and the firm could offer almost all of its products. To better evaluate algorithms, we consider different capacities by multiplying a scale factor α to the capacity of legs c. We use α = 0.6, 0.8, 1, 1.5 and 2 to solve the problem and the booking horizon consists of 1500 periods of time. Recall that at each period of time, at most one customer arrives and requests one unit of the products. As mentioned in the previous chapter, we suppose that each day consists of approximately 140 periods of time and hence we consider every 10 minutes to be a unit of time t. Furthermore, we make a relaxation on variables $t(S)$ and we suppose that they are continuous variables (*i.e.* we can offer a set of products for a fraction of time as well).

Computational results for the small network example

The results obtained by implementing different solution approaches are presented in Table (4.3). The first column represents the scale factor α used to change the initial capacity of legs. The rest of the table shows the final number of variables in the master problem (FN), the average number of column generation iterations (AI), the average computational time in seconds, the revenue and the average maximum number of nodes in the branch and bound processing, respectively. Note that we consider an average of ten instances for each scenario.

Studying the obtained results, we can see that in the cases α = 0.6, 0.8 and 2, we obtain an optimal solution by using the Dinkelbach-based heuristic to solve the column generation subproblem, while by implementing the greedy heuristic we could find a solution with a small optimality gap. In the cases α = 1 and 1.5, there is a small gap by implementing the Dinkelbach-based heuristic compared to the greedy heuristic. Note that as we are considering a small network, there are not notable differences in the revenue and solution time. Even though there are just 22 products available in the definition of the problem, the firm could choose between $2^{22} - 1$ possible non-empty sets of products to offer to the

Table 4.3 Results of solution approaches for the small airline network problem.

α	Solution approach	FN	AI	Time(sec.)	Revenue	AN
0.6	MIP	*42*	*40*	*7.8*	*221758*	*140*
	Greedy heuristic	42	40	5.7	221566	-
	Dinkelbach-Based Heuristic	**43**	**41**	**6.5**	**221758**	-
0.8	MIP	*38*	*36*	*7.4*	*280304*	*140*
	Greedy heuristic	34	32	4.6	279533	-
	Dinkelbach-Based Heuristic	**36**	**34**	**5.4**	**280304**	-
1	MIP	*30*	*28*	*9.4*	*315921*	*648*
	Greedy heuristic	26	24	3.8	315882	-
	Dinkelbach-Based Heuristic	**27**	**25**	**4.1**	**315913**	-
1.5	MIP	*19*	*17*	*19.7*	*352926*	*3800*
	Greedy heuristic	22	20	3.1	352824	-
	Dinkelbach-Based Heuristic	**14**	**12**	**1.3**	**352874**	-
2	MIP	*8*	*6*	*1.2*	*357844*	*230*
	Greedy heuristic	8	6	0.4	357341	-
	Dinkelbach-Based Heuristic	**8**	**6**	**0.4**	**357844**	-

customers. This means that it is not possible to solve the CDLP model directly with this amount of variables. Nonetheless, column generation, as shown by the results of Table (4.3) helps us to find an appropriate solution with just a few iterations and variables.

Regarding the results reported in Table (4.3), another approach to find the solution for the airline network could be combining exact and heuristic methods together to solve the column generation subproblem. This means that first we start by using our heuristic approach to solve the subproblem and find new variables to enter into the master problem. Afterwards, when the heuristic approach fails to find a column with a positive reduced cost, we use the MIP method to check if there are columns with a positive reduced cost to enter the problem. These steps continue until both algorithms fail to find a new entering column. The latter means that the solution is optimal. Regarding the results in Table (4.3), with the Dinkelbach-based algorithm, in most of the cases, the optimal solution could be found without needing to use the exact method, but by using the greedy heuristic, we need using the exact algorithm more often. As we know, because of the complexity of the exact method, this hybrid method can be applied just on small and medium size network problems.

Another interesting fact according to the results of Table (4.3) is the effect of the capacity on the number of iterations (*i.e.* entering columns). As is shown in Figure (4.2), we see that by increasing the capacity of legs, the complexity of the problem decreases, and because of the lower complexity, the problem can be solved with a fewer number of iterations. This is because by increasing capacity, the firm can fulfill a larger amount of customers' demands. In the case in which the firm faces higher capacity on legs than customers demand, the problem becomes easy to solve and the best policy could be to offer almost all of the products to the customers.

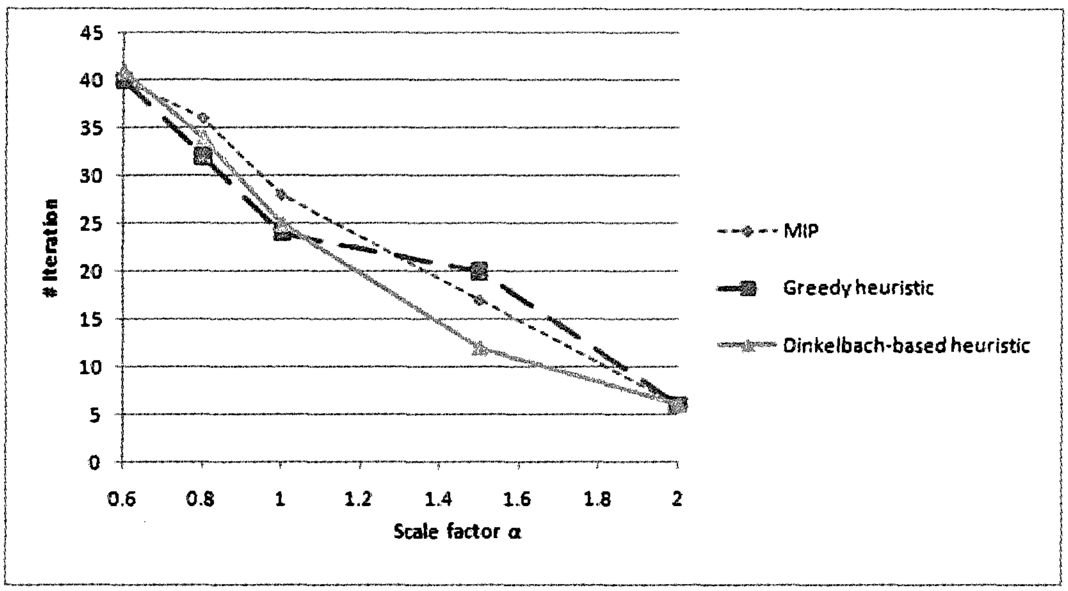

Figure 4.2 Effect of scale factor α on the number of iterations on different solution approaches

4.2 Thalys railroads example

This example is based on Thalys railroads, of which we will consider a part of its network with five cities and four legs. There are two high (H) and low (L) fare classes on each leg. Figure (4.3) illustrates the Thalys railroad network and its associated market.

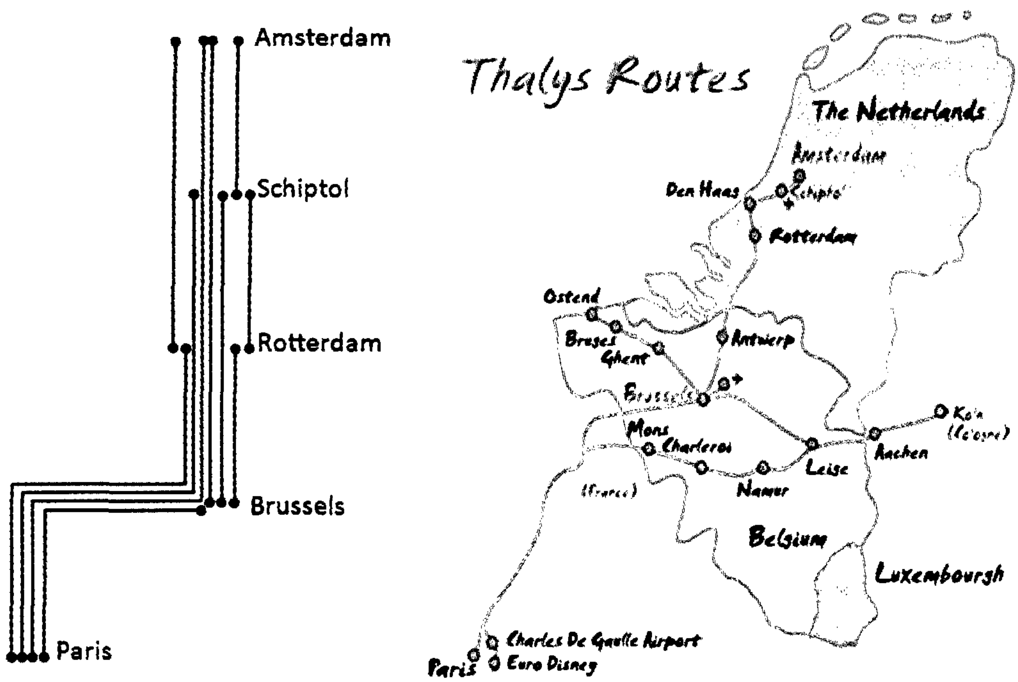

Figure 4.3 Thalys railroad network and its associated market

In this problem, there are 10 trains with a capacity of 100 passengers going from Paris to Amsterdam. Each train stops in Brussels, Rotterdam, Schiptol, and Amsterdam. Thus, there are 10 markets shown in Figure (4.3). Two fare classes and 10 markets produce a total of 200 products (*i.e.* train fare combinations). Table (4.4) shows price information associated with each market.

Table 4.4 Markets and their relative prices

Market	Low fare	High fare
PAR → BRU	200	400
PAR → RTA	300	500
PAR → SCH	350	525
PAR → AMA	350	525
BRU → RTA	150	250
BRU → SCH	175	275
BRU → AMA	200	300
RTA → SCH	50	100
RTA → AMA	175	300
SCH → AMA	50	100

We divide customers into 20 different segments based on their sensitivity to price and their origin and ultimate destination. Price sensitive (leisure) customers prefer low fare, but they can still buy high fare as well, while price insensitive (business) customers only choose high fares. Table (4.5) shows each segment's definition according to our assumptions.

We solve the problem in different booking horizons. Recall that at each period of time, at most one customer arrives and requests one unit of the products. As mentioned in the previous chapter, we suppose that each day consists of approximately 140 periods of time and hence we consider every 10 minutes to be a unit of time t.

Table 4.5 Segment definition for Thalys railroad network

Sgm.	Consideration	Preference vector	Description
1	{1,...,20}	{10,55,25,15,6,4,3,4,5,6,15,15,20,4,3,2,1,2,2,3,8}	PAR-BRU (L)
2	{11,...,20}	{8,70,60,10,7,4,4,4,5,40,60}	PAR-BRU (H)
3	{21,...,40}	{15,30,20,10,3,5,20,25,10,4,4,4,8,2,1,2,2,3,3,2,2}	PAR-RTA (L)
4	{31,...,40}	{7,40,25,10,4,4,5,15,20,25,45}	PAR-RTA (H)
5	{41,...,60}	{25,25,20,4,5,5,5,6,6,10,30,5,2,2,2,3,3,3,4,4,10}	PAR-SCH (L)
6	{51,...,60}	{7,32,21,3,3,4,5,15,15,20,30,}	PAR-SCH (H)
7	{61,...,80}	{20,20,2,5,5,6,6,7,7,8,15,3,3,4,3,3,4,4,5,4,4}	PAR-AMA (L)
8	{71,...,80}	{50,25,20,3,3,4,4,8,20,28,35}	PAR-AMA (H)
9	{81,...,100}	{10,60,50,6,4,4,5,20,22,7,32,10,4,3,2,2,2,3,4,4,15}	BRU-RTA (L)
10	{91,...,100}	{20,90,45,5,6,2,3,4,30,60,70}	BRU-RTA (H)
11	{101,...,120}	{5,25,10,5,5,6,6,20,20,10,8,5,4,3,3,3,4,4,5,5,5}	BRU-SCH (L)
12	{111,...,120}	{10,35,7,6,4,4,5,6,7,35,40}	BRU-SCH (H)
13	{121,...,140}	{30,24,4,4,3,3,5,6,6,10,10,3,2,2,2,2,3,4,5,5,6}	BRU-AMA (L)
14	{131,...,140}	{15,8,6,5,4,5,6,7,10,12,10}	BRU-AMA (H)
15	{141,...,160}	{10,25,20,4,4,3,3,4,5,6,10,4,4,3,2,2,3,3,4,4,4}	RTA-SCH (L)
16	{151,...,160}	{4,34,36,3,2,2,4,4,5,25,30}	RTA-SCH (H)
17	{161,...,180}	{20,40,10,5,4,3,4,5,5,6,25,4,2,1,2,2,2,3,4,4,5}	PAR-AMA (L)
18	{171,...,180}	{5,50,25,25,3,4,5,6,6,35,40}	RTA-AMA (H)
19	{181,...,200}	{30,32,20,5,4,4,4,5,6,7,20,4,4,3,2,3,3,4,4,5,5}	SCA-AMA (L)
20	{191,...,200}	{15,40,20,4,4,4,5,6,6,35,60}	SCA-AMA (H)

Like the previous example, we alter the capacity of legs by multiplying a scale factor α to the capacity of legs c. The experiments are done for three different $\alpha = 0.5$, 1, and 1.5.

Computational results for α = 0.5 over different booking horizons

In this section, we consider the Thalys railroad example for $\alpha = 0.5$ and hence $c = 50$. We evaluate these three approaches by changing the number of periods from 100 to 3000. Table (4.6) summarizes the results obtained under different policies.

As before, we use NP as number of periods, FN as Final number of variables, AI as average number of iterations, AT as average cpu time in seconds and AN as average number of maximum number of nodes. Note that initial columns are selected randomly.

Table 4.6 Results for different solution approaches considering $\alpha = 0.5$

NP	Solution approach	FN	AI	AT	Revenue	AN
T=100	MIP	8	4	6593	33365.3	177754
	Greedy Heuristic	8	4	91	33365.1	-
	Dinkelbach-Based Heuristic	**8**	**4**	**26**	**33365.3**	**-**
T=500	MIP	20	16	115200	159797	1344200
	Greedy Heuristic	24	20	994	159781	-
	Dinkelbach-Based Heuristic	**21**	**17**	**402**	**159797**	**-**
T=1000	MIP*	-	-	184700	282879	2503500
	Greedy Heuristic	111	107	19803	285425	-
	Dinkelbach-Based Heuristic	**109**	**105**	**16492**	**286224**	**-**
T=2000	MIP	-	-	-	-	-
	Greedy Heuristic	180	176	23218	407336	-
	Dinkelbach-Based Heuristic	**175**	**171**	**21538**	**407738**	**-**
T=3000	MIP	-	-	-	-	-
	Greedy Heuristic	202	198	35621	452403	-
	Dinkelbach-Based Heuristic	**191**	**187**	**31085**	**455757**	**-**

In the case $T = 100$ periods, we see that all methods could find an optimal solution and the number of iterations are the same in all of them. We also observe the effect of the complexity of exact method on the cpu time, resulting in a large difference in the time used for the exact method compared to the greedy or the Dinkelbach-based heuristic. We can see that the time used for the alternative methods is almost less than 0.01 percent of the exact method.

In the case $T = 500$ periods, the results show different numbers of iterations for the different approaches. The Dinkelbach-based heuristic could find the same revenue as the exact method with just one more iteration and the greedy one finds an answer near to the optimal revenue but with more time spent and more iterations. It can be observed clearly from the computation time that there is a significant difference in processing time between the exact and heuristic methods in this case.

If we increase the size of the problem, the impact of the complexity will be much clearer. When we increase T to 1000 periods, because of the size of problem and without limiting the search in the exact algorithm, MIP after more than 2 days running stops at a revenue worse than the amount we obtain after less than 5 hours with the Dinkelbach-Based heuristic. The greedy one also works much better than the exact algorithm, but with less efficiency according to the new heuristic where the greedy one has 20 percent more computing time.

By increasing the booking horizon to 2000, and even more to 3000 periods, we just compare two alternative methods. It can be observed that for $T = 2000$ periods, both algorithms still find close revenues, while for $T = 3000$ periods, the difference in revenue becomes more than the previous cases. Regarding the processing time for the greedy heuristic, there is at least a 15 percent improvement by implementing the new heuristic approach.

Computational results for $\alpha = 1$ and $\alpha = 1.5$ on different booking horizons

In this section, we consider the Thalys railroad example for $\alpha = 1$ and $\alpha = 1.5$ and hence $c = 100$ and $c = 150$. The number of periods are varying from 100 to 3000 periods of time. Unlike the results obtained by $\alpha = 0.5$, we evaluate only two heuristic approaches (*i.e.* the greedy heuristic and Dinkelbach-based heuristic).

Tables (4.7) and (4.8) summarize the results obtained under different policies. We use NP as number of periods, FN as Final number of variables, AI as average number of iterations, VP as number of variables with the positive value in the base, OP as the number of offering products to the customers, AT as average cpu time in seconds and AN as average number of maximum number of nodes

Table 4.7 Results for different solution approaches considering $\alpha = 1$

NP	Solution approach	FN	AI	VP	OP	AT	Revenue
T=100	Greedy h.	7	3	1	70	63	33365.1
	Dinkelbach-based h.	7	3	1	70	17	33365.3
T=200	Greedy h.	7	3	1	70	63	66730.2
	Dinkelbach-based h.	7	3	1	70	17	66730.2
T=500	Greedy h.	14	10	2	72	272	166163
	Dinkelbach-based h.	12	8	2	73	80	166163
T=1000	Greedy h.	23	19	5	83	700	319562
	Dinkelbach-based h.	20	16	5	83	283	319595
T=1500	Greedy h.	61	57	10	93	4233	454166
	Dinkelbach-based h.	40	36	10	94	1552	455224
T=2000	Greedy h.	110	106	15	107	12923	570850
	Dinkelbach-based h.	104	100	16	103	9558	572447
T=3000	Greedy h.	182	178	25	86	25549	727754
	Dinkelbach-based h.	148	144	24	88	15122	727776

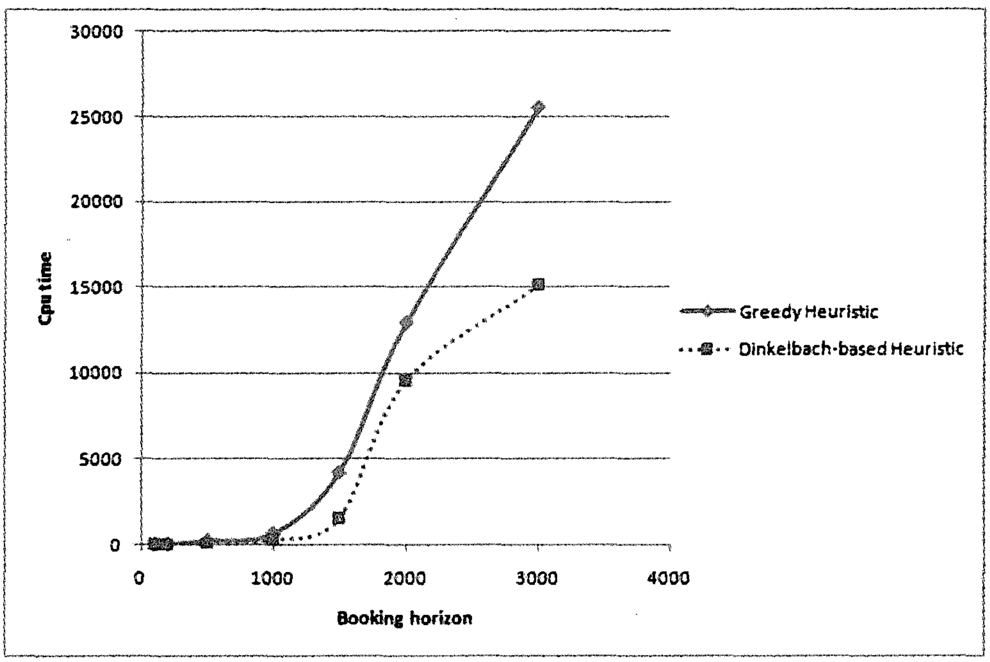

Figure 4.4 CPU time comparison between two heuristic approaches $\alpha = 1$

Table 4.8 Results for different solution approaches considering $\alpha = 1.5$

NP	Solution approach	FN	AI	VP	OP	AT	Revenue
T=100	Greedy h.	7	3	1	70	64	33365.1
	Dinkelbach-based h.	7	3	1	71	17	33365.3
T=200	Greedy h.	7	3	1	70	64	66730
	Dinkelbach-based h.	7	3	1	71	17	66730.6
T=500	Greedy h.	14	10	2	72	272	166163
	Dinkelbach-based h.	7	3	1	71	17	166827
T=1000	Greedy h.	15	11	3	74	345	330053
	Dinkelbach-based h.	12	9	3	74	111	330053
T=1500	Greedy h.	23	19	5	83	835	479343
	Dinkelbach-based h.	20	16	5	83	318	479392
T=2000	Greedy h.	38	34	8	89	2078	616783
	Dinkelbach-based h.	35	31	8	93	1187	617654
T=3000	Greedy h.	110	106	15	108	13723	856275
	Dinkelbach-based h.	103	99	16	102	10897	858671
T=4000	Greedy h.	188	184	22	102	31265	1030420
	Dinkelbach-based h.	144	140	22	100	18427	1030750

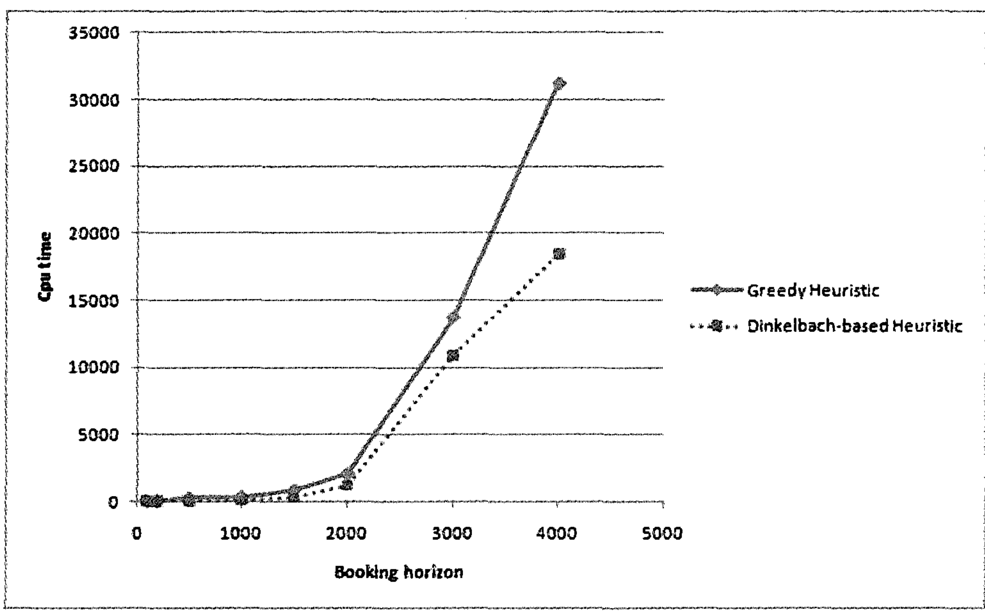

Figure 4.5 CPU time comparison between two heuristic approaches $\alpha = 1.5$

Figures (4.4) and (4.5) illustrate the difference of these two algorithms. It can be observed that as much as the size of the problem grows, the efficiency of the Dinkelbach-based heuristic is going to be much better than the greedy heuristic. Also, by considering the revenue obtained in the different booking horizons shown in Tables (4.7) and (4.8), either in terms of time consumed for the processing or in terms of the quality of the solutions, both indicate that it is more reasonable to use the Dinkelbach-based heuristic instead of the greedy one for practical and real size networks.

We illustrate more details of the column generation algorithm's steps in Figure (4.6). The left and right columns of this figure represent the first five steps of this algorithm using the Dinkelbach-based heuristic or greedy heuristic, respectively. The amount of time spent in the master phase of the algorithm and in the subproblem are mentioned in each iteration as well.

```
***************************************************************
Revenue = 75108.2

time spend for iteration 1 in Master phase = 1.714
time spend for iteration 1 in Dinkel-based heuristic = 0.057
Full time spend till iteration 1 is = 1.78
***************************************************************
Revenue = 257934

time spend for iteration 2 in Master phase = 4.02
time spend for iteration 2 in Dinkel-based heuristic = 0.031
Full time spend till iteration 2 is = 5.831
***************************************************************
Revenue = 407444

time spend for iteration 3 in Master phase = 5.89
time spend for iteration 3 in Dinkel-based heuristic = 0.057
Full time spend till iteration 3 is = 11.778
***************************************************************
Revenue = 447004

time spend for iteration 4 in Master phase = 7.782
time spend for iteration 4 in Dinkel-based heuristic = 0.029
Full time spend till iteration 4 is = 19.589
***************************************************************
Revenue = 598729

time spend for iteration 5 in Master phase = 9.544
time spend for iteration 5 in Dinkel-based heuristic = 0.03
Full time spend till iteration 5 is = 29.163
***************************************************************
```

```
***************************************************************
Revenue = 75108.2

time spend for iteration 1 in Master phase = 1.602
time spend for iteration 1 in Greedy heuristic = 17.011
Full time spend till iteration 1 is = 18.616
***************************************************************
Revenue = 257812

time spend for iteration 2 in Master phase = 3.428
time spend for iteration 2 in Greedy heuristic = 16
Full time spend till iteration 2 is = 38.044
***************************************************************
Revenue = 397641

time spend for iteration 3 in Master phase = 5.171
time spend for iteration 3 in Greedy heuristic = 15.67
Full time spend till iteration 3 is = 58.885
***************************************************************
Revenue = 441862

time spend for iteration 4 in Master phase = 6.951
time spend for iteration 4 in Greedy heuristic = 13.637
Full time spend till iteration 4 is = 79.474
***************************************************************
Revenue = 593096

time spend for iteration 5 in Master phase = 8.533
time spend for iteration 5 in Greedy heuristic = 14.782
Full time spend till iteration 5 is = 102.789
***************************************************************
```

Figure 4.6 Comparison of the first 5 steps of the column generation algorithm for $\alpha = 1.5$, and $T = 4000$, implementing two heuristics approaches

These results also show that in each step of the Dinkelbach-based heuristic, we have a better revenue and computation time when compared to the greedy heuristic. The average elapsed time in the greedy heuristic is approximatively 15 seconds while we have only 0.05 seconds in the average elapsed time for solving the subproblem by using the Dinkelbach-Based Heuristic, which is a significant improvement in the ultimate computation time.

The graph in figure (4.7) compares the speed of the column generation algorithm for the case with $\alpha = 1$ and $T = 500$ periods, where we want to reach the ultimate solution by using the two different approaches mentioned to solve the subproblem. This graph obviously indicates the better speed of the Dinkelbach-based heuristic. As it can be easily observed, by using this approach, the column generation algorithm converges to its ultimate solution much faster than using the greedy one.

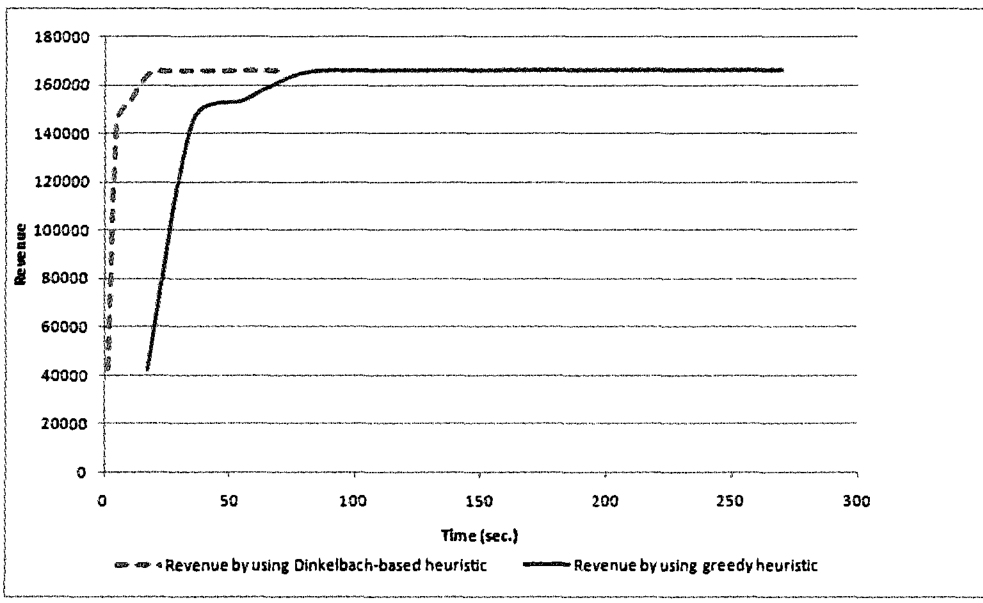

Figure 4.7 Comparison of two different approaches for solving the subproblem on the speed of column generation algorithm ($\alpha = 1$ and $T = 500$)

The Thalys railroad example also shows the effect of altering capacity in processing time and obtained revenue. As we expected, the problem becomes easier to solve by increasing available capacity to offer the products to customers, so the firm can offer most of its products. The other important fact that should be mentioned with regards to the results obtained in Tables (4.6), (4.7) and (4.8) is the small number of iterations compared to the original model's size.

In our problem, despite the fact that the original model has $2^{200}-1$ variables, in the most difficult case that we considered, the column generation algorithm reaches the solution in a maximum of 187 iterations by using the Dinkelbach-based algorithm. The number of iterations by using this approach is also less than the number of iterations done by implementing the greedy heuristic.

As mentioned in the small airline's example, one approach to solve customer choice-based models could be mixing heuristic and exact methods. To do so, we start by a heuristic approach to find rapidly entering columns until the heuristic fails to find any column with a positive reduced cost. At this time, we use the exact method to find new solutions.

Because of the fact that processing time and the quality of the solution are two main aspects in the practical models, and because of the complexity issue of this model, here it is illogical to use MIP method on real-size networks like the Thalys railroads network. Furthermore, according to the obtained results, using only a heuristic method does not cause a meaningful loss of revenue (if there was any). Therefore, using the heuristic without an exact method seems to be more reasonable for the practical real size networks.

Comparing the greedy heuristic and Dinkelbach-based heuristic with single tests

In this section, we evaluate the two heuristic approaches by simply implementing them on single tests of sum of ratios problems, instead of comparing results done by applying them in several iterations of the column generation algorithm. To do so, we randomly generate sum of ratios problems with a structure similar to our subproblem, and subsequently, solving them with a different number of ratios and variables, we compare the efficiency of these two algorithms. Table (4.9) presents the average results of ten different executions on each considered case. Note that a positive average gap represents a better solution for the Dinkelbach-based heuristic.

Table 4.9 Analysis of two heuristic methods solving randomly generated sum of ratios problems

Instance		Average	Average time (Sec.)	
# Ratios	#variables	gap %	Greedy heuristic	Dinkelbach-based heuristic
10	50	0.02	3	0.01
10	100	-0.58	10	0.01
10	200	1.87	31	0.02
10	500	0.02	356	0.04
10	1000	0.01	1918	0.1
20	50	0.02	5	0.01
20	100	-0.01	18	0.01
20	200	1.03	60	0.05
20	500	0.02	600	0.08
30	50	0.02	7	0.01
30	100	0.02	35	0.03
30	200	0.1	92	0.06
30	500	0.05	1430	0.1
50	50	0.02	11	0.03
50	100	0.03	49	0.05
50	200	0.24	100	0.1
50	500	0.03	2328	0.4

These obtained results make the advantage of using the Dinkelbach-based heuristic clearer. Even though there is not a significant difference in the solutions obtained by two approaches, there is a meaningful improvement in reducing computational time by using the second heuristic.

The results indicate that increasing the number of variables while the number of ratios is fixed causes an increase of computational time by using the greedy heuristic, while the time spent in the second algorithm is too short compared to the greedy one. These results could be easily observed in the table.

Similarly, by increasing the number of ratios while the number of variables are fixed, an increase in computational time can be observed. The latter is shown in Figures (4.8) and (4.9). Based on the results of Table (4.9) and fixing the number of variables to 500 variables, these figures show an increasing trend in the average time with respect to the number of ratios. Note that the average time in the Dinkelbach-based heuristic is much smaller than in the greedy one because instead of doing lots of time consuming comparisons in the greedy heuristic in the new approach we just solve a series of LP problems.

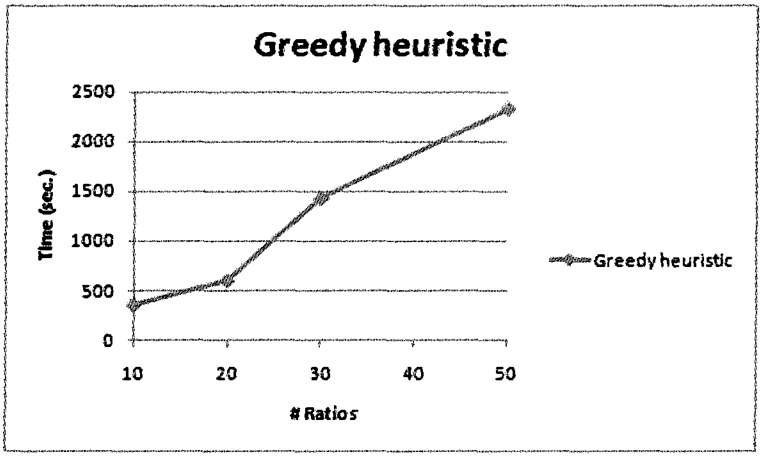

Figure 4.8 Processing time as a function of number of ratios by using the greedy heuristic (500 variables)

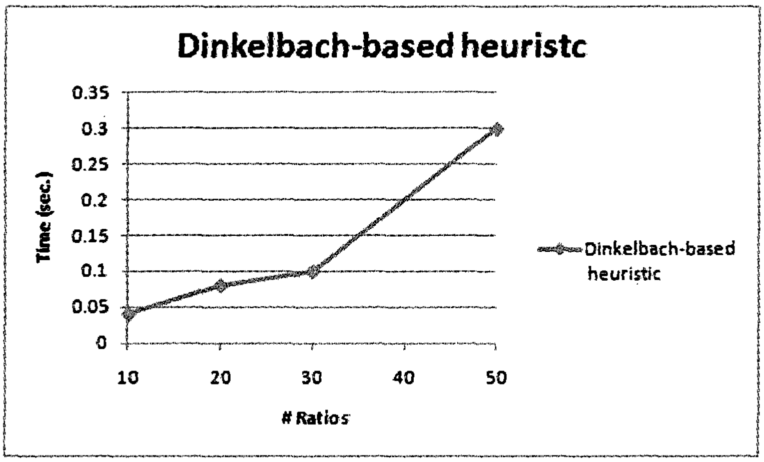

Figure 4.9 Processing time as a function of number of ratios by using the Dinkelbach-based heuristic (500 variables)

CHAPTER 5

CONCLUSION AND DISCUSSION

Taking into account customer choice behavior can be considered a departure from traditional revenue management methodologies. While most traditional models can not implement consumer behaviors such as buy-up or buy-down, or at best they use just some approximate heuristics to model these behaviors, using customer choice-based revenue management models presents a significant improvement to the firm's decision making process.

In this research, we considered two main challenges that we face when implementing a choice-based revenue management model. First, we reviewed the modeling of customer choice behaviors and estimating choice behavior from available data. Second, we investigated revenue optimization techniques that can deal with complex, choice-based models of demand.

In this study, we considered the choice-based, deterministic, linear programming (CDLP) model of Gallego et al. [20] and further works done by Van Ryzin and Liu [40] and Vulcano [9] in which customers can belong to more than one segment, according to a multinomial logit model. Regarding the exponential number of variables of the CDLP model for these real-world practical problems, a column generation algorithm is developed to solve this large-scale optimization problem.

However, using the column generation algorithm, we faced another challenge. Indeed, the column generation subproblem formulated as a 0-1 linear fractional programming problem with the summation of several ratios is proven to be NP-hard. To tackle this complexity, several solution approaches were studied. First, we started by an exact method. Using some linearization techniques, a mixed integer programming was obtained.

However, considering the complexity of the problem, some alternative approaches should be used to solve the problem in polynomial time. We continued by studying the greedy heuristic approach proposed by Vulcano et al. [9] and afterwards we proposed a new heuristic approach to tackle this problem. According to our computational results, the new heuristic has a noteworthy performance and in terms of the quality of the obtained solution and processing time, it performed better than the greedy heuristic.

There are several topics that would be worth considering for further works. One is improving the column generation algorithm in such a way that instead of a single column, we would have more useful entering columns in the master problem. Simultaneously, in order to achieve higher performance, improving heuristic approaches could be useful as well.

Another worthwhile extension of the algorithm could be supposing uncertain preference vectors for customers. That means the probability of choosing a certain product could be changed during the booking horizon and leads us to have a better interpretation of consumer behavior and hence better decision-making policies. Moreover, studying dynamic programming decomposition approaches could also be an interesting approach to improve available procedures.

REFERENCES

[1] Almogy, Y. and Levin, O. (1971). A class of fractional programming problems. *Operations Research* vol. 19, pp. 57-67.

[2] Almogy, Y. and Levin, O. (1970). Parametric analysis of a multi-stage stochastic shipping problem. Proc Fifth IFORS Conf, Venice. pp. 359-370.

[3] Bajalinove, E.B. (2004). Linear Fractional Programming. *Kluwer Academic Publishers*, Boston.

[4] Belobaba, P. P. and Hopperstad, C. (1999). Boeing/MIT Simulation Study: PODS Results Update. presented at AGIFORS Reservations and Yield Management Study Group Symposium Proceedings, London, UK.

[5] Belobaba, P. P. (1987). Airline Yield Management: An Overview of Seat Inventory Control. *Transportation Science*, vol. 21, pp. 63-73.

[6] Ben-Akiva M. and Lerman. S. *Discrete Choice Analysis: Theory and Applications to Travel Demand.* The MIT Press, Cambridge, MA, sixth edition, 1994.

[7] Benson H.P. (2007). Solving sum of ratios fractional programs via concave minimization. *Journal of Optimization Theory and Applications*, vol. 135(1), pp. 117.

[8] Benson, H.P. (2001). Global optimization of nonlinear sums of ratios. *Journal of Mathematical Analysis and Applications* vol. 263, pp. 301-315.

[9] Bront, J. J. M. I., Mendez-Diaz, and Vulcano, G. (2007). A Column Generation Algorithm for Choice-Based Network Management. pp. 25, Paper Submitted to AGIFORS for Anna Valicek Medal 2007.

[10] Cravan, B.D. (1988). Fractional Programming, Sigma series in Applied Mathematics. Heldermann, Berlin.

[11] Chen, D.Z. Daescu, O. Dai, Y. Katoh, N. Wu, X. and Xu, J. (2000). Optimizing the sum of linear fractional functions and applications. *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 707-716, ACM, New York.

[12] Cross, R. G. (1997). Revenue Management: Hard-Core Tactics for Market Domination, *Broadway Books*. New-York.

[13] Crouzeix, J. and Ferland, J. (1991). Algorithms for generalized fractional programming. *Mathematical Programming*, vol. 52(2), pp. 191-207.

[14] Crouzeix, J., Ferland, J., and Schaible, S. (1986). A note on an algorithm for generalized fractional programs. *Journal of Optimization Theory and Applications*, vol. 50(1), pp. 183-187.

[15] Dash Associates. (1999). *XPRESS-MP Reference Manual.*

[16] Drezner, Z. Schaible, S. and Simchi-Levi. D. (1990). Queueing-location problems on the plane. *Naval Research Logistics* vol. 37, pp. 929-935.

[17] Falk, J.E. and Palocsay, S.W. (1992). Optimizing the sum of linear fractional functions, in C.A. Floudas and P.M. Pardalos (eds.). *Recent Advances in Global Optimization*, pp. 221-258, Kluwer Academic Publishers, Dordrecht.

[18] Falk, J.E. and Palocsay, S.W. (1994). Image space analysis of generalized fractional programs. *Journal of Global Optimization* vol. 4 , pp. 63-88.

[19] Freund, R.W. and Jarre, F. (2001). Solving the sum-of-ratios problem by an interior point method. *Journal of Global Optimization* vol. 19, pp. 83-102.

[20] Gallego, G., Iyengar, G., Phillips, R., and Dubey. A. (2004). Managing flexible products on a network. Technical Report CORC TR-2004-01. Department of Industrial Engineering and Operations Research, Columbia University.

[21] Hodgson, T.J. and Lowe, T.J. (1982). Production lot sizing with material handing cost considerations. *IIE Transactions* vol. 14, pp. 44-51.

[22] Konno, H. and Fukaishi, K. (2000). A branch and bound algorithm for solving low rank linear multiplicative and fractional programming problems. *Journal of Global Optimization* vol. 18, pp. 283-299.

[23] Konno, H. and Inori, M. (1989). Bond portfolio optimization by bilinear fractional programming. *Journal of the Operations Research Society of Japan* vol. 32, pp. 143-158.

[24] Konno, H. and Yamashita, H. (1999). Minimization of the sum and the product of several linear fractional functions. *Naval Research Logistics* vol. 46, pp. 583-596.

[25] Kuno, T. (2002). A branch-and-bound algorithm for maximizing the sum of several linear ratios. *Journal of Global Optimization* vol. 22, pp. 155-174.

[26] Littlewood, K. (1972). Forecasting and control of passenger bookings. AGIFORS Sympos. Proc. vol. 12, pp. 95-117.

[27] Martos, B. (1964). Hyperbolic Programming. *Naval Res. Log. Quart.* vol. 11, pp. 135-155.

[28] Mathis, F.H. and Mathis, L.J. (1995). A nonlinear programming algorithm for hospital management. *SIAM Review* vol. 37, pp. 230-234.

[29] Matsui, T. , Saruwatari, Y. Maiko Shigeno. (1992). An Analysis of Dinkelbach's Algorithm for 0-1 Fractional Programming Problems.

[30] Prokopyev, O., Huang, H., and Pardalos, P. (2005). On complexity of unconstrained hyperbolic 0-1 programming problems. *Operations Research Letters* vol. 33, pp. 312-318.

[31] Rao, M.R. (1971). Cluster analysis and mathematical programming. *Journal of the American Statistical Association*, vol. 66, pp. 622-626.

[32] Schaible, S. (2001). Fractional programming - a recent survey, *Journal of Statistics and Management Systems*, vol. 29, pp. 845-866.

[33] Schaible, S. and Lowe, T. (1983). A note on a material control problem. *IIE Transactions* vol. 15, pp. 177-179.

[34] Schaible, S., and Shi, J.(2002). Fractional Programming: The Sum-of-Ratios Case. Working Paper 02-13. Graduate School of Management, University of California, Riverside, California.

[35] Scott, C. H. and Jefferson, T. R. (1998). Duality of a nonconvex sum of ratios. *Journal of Optimization Theory and Applications* vol. 98, pp. 151-159.

[36] Smith, B., Leimkuhler, J. , and Darrow, R. (1992). Yield Management at American Airlines. *Interfaces*, vol. 22, pp. 8-31.

[37] Talluri, K. and van Ryzin, G. (2004). The Theory and Practice of Revenue Management. *Kluwer Academic Publishers*, New York, NY.

[38] Talluri, K. T., and van Ryzin, G. J. (2004). Revenue management under a general discrete choice model of consumer behavior. *Management Science* vol. 50 , pp. 15-33.

[39] Van Ryzin, G. (2005). Models of Demand. *Journal of Revenue & Pricing Management*, vol. 4, pp. 204-210.

[40] Van Ryzin, G., and Liu, Q. (2004). On the choice-based linear programming model for network revenue management. Working paper, Graduate School of Business, Columbia University, New York.

[41] Vulcano, G., van Ryzin, G. J. and Chaar, W. (2009). Choice-Based Revenue Management: An Empirical Study of Estimation and Optimization. Working paper available at http://pages.stern.nyu.edu/gvulcano/Research.html.

[42] Wei-Ying W. , Ruey-Lin Sheu , S. Ilker Birbil. (2008). Solving the sum-of-ratios problem by a stochastic search algorithm. *Journal of Global Optimization*, vol. 42 n.1, pp. 91-109.

[43] Wu, T. (1997). A note on a global approach for general 0-1 fractional programming. *European Journal of Operational Research* vol. 101, pp. 220-223.