Computers & Operations Research 69 (2016) 25–39

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Efficient heuristic algorithms for maximum utility product pricing problems

T.G.J. Myklebust [1], M.A. Sharpe [2], L. Tunçel [3,*]

Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1

A R T I C L E I N F O

Available online 17 December 2015

AMS Subject Classification::
90C11
91B25
90C27
90C35
68C05
05C85

Keywords:
Mixed integer programming
Network flow problems
Total unimodularity
Optimal product pricing
Algorithms

A B S T R A C T

We propose improvements to some of the best heuristic algorithms for optimal product pricing problem originally designed by Dobson and Kalish in the late 1980s and in the early 1990s. Our improvements are based on a detailed study of a fundamental decoupling structure of the underlying mixed integer programming (MIP) problem and on incorporating more recent ideas, some from the theoretical computer science literature, for closely related problems. We provide very efficient implementations of the algorithms of Dobson and Kalish as well as our new algorithms. We show that our improvements lead to algorithms which generate solutions with better objective values and that are more robust in overall performance. Our computational experiments indicate that our implementation of Dobson–Kalish heuristics and our new algorithms can provide good solutions for the underlying MIP problems where the typical LP relaxations would have more than a trillion variables and more than three trillion constraints.

1. Introduction

We study a class of optimal product pricing problems in which the customer demands, budgets and preferences are encoded by *reservation prices* − −the highest price that a customer is willing and able to pay for a given product. See, for instance, [18–20]. Essentially every company operating in a free market environment faces a variant of this problem. Therefore, the optimal pricing problem in revenue management is widespread. A very fundamental version of the problem is based on the assumption that the customers will buy the product that maximizes their individual surplus (or utility), where the surplus is defined as the difference between the reservation price of the customer for the product and the price of the product. In some similar contexts, this model can also be used for *envy−free pricing*; see the related models in [1,3,11]. Maximum utility pricing models are also related to bilevel pricing problems. For the latter, see, for instance, [15] and the references therein. The problem considered there is equivalent to maximum utility product pricing problem, see [16].

From a computational complexity viewpoint, the problem is $\mathcal{NP}$−hard, see [9,3]; it is $\mathcal{NP}$−hard even to approximate within a reasonable worst−case ratio, see [11] (even though some special cases of related optimal pricing problems admit efficient algorithms, see for instance [12]). Currently, the best heuristic algorithms for solving these problems are the ones proposed by Dobson and Kalish, see [8,9]. While these algorithms are successful on many instances of the problem, quite often they are not.

We present very efficient implementations of Dobson–Kalish heuristic algorithms for optimal pricing problems [8,9]. Then, we consider further improvements based on a decoupling property of the problem with respect to continuous and binary variables. Our improvements are based on a detailed study of a fundamental decoupling structure of the underlying mixed integer programming (MIP) problem and on incorporating some more recent ideas for closely related problems from Computer Science, Operations Research and Optimization. We provide very efficient implementations of our new algorithms; efficiency, in particular the speed of the algorithms is critical in practical applications of our framework. A main reason is our stationary price assumption for the competitors' products. Indeed, in practice the competitors may adjust their prices, as a result, we should be able to re−optimize

* Corresponding author. Tel.: +1 519 888 4567x35598; fax: +1 519 725 5441.
E-mail addresses: tmyklebu@csclub.uwaterloo.ca (T.G.J. Myklebust), masharpe@uwaterloo.ca (M.A. Sharpe), ltuncel@uwaterloo.ca (L. Tunçel).

[1] Research of this author was supported in part by a Tutte Scholarship, and Discovery Grants from NSERC.

[2] Research of this author was supported in part by a Summer Undergraduate Research Assistantship Award from NSERC (Summer 2009) and a Discovery grant from NSERC.

[3] Research of this author was supported in part by Discovery Grants from NSERC and by research grants from University of Waterloo.

http://dx.doi.org/10.1016/j.cor.2015.11.013
0305-0548/

our prices and respond very quickly. Another main reason for the required speed comes from the sizes of the optimization problems in nontrivial applications. For the sake of presentation, we make simplifying assumptions about the optimal pricing problem while constructing the mathematical model. Indeed, in applications, we can relax most of the assumptions required by our mathematical models. However, to do so, we introduce new variables to the problem. For example, the mathematical model requires that every customer segment pay the same price for the same product. Suppose that in some application, we would like to offer different prices to customers who buy their plane tickets four to six weeks in advance of their trip than others. In this case, we simply create a new product in our mathematical model to represent these plane tickets. Another instance arises in product bundling. For each bundle of products, we create a new variable which represents the bundle. These transformations when properly applied can lead to tens of thousands of "product variables" in the mathematical model (which in turn lead to LP relaxations with a huge number of variables and constraints).

We show that our improvements yield algorithms which generate solutions with better objective values and that are more robust in overall performance. Our computational experiments indicate that our implementation of Dobson–Kalish heuristics and our new algorithms can provide good solutions for the underlying MIP problems where the typical LP relaxations would have more than a trillion variables and more than three trillion constraints. In the next section, we begin the development of the necessary notation and the description of some of the fundamental mathematical models we utilize. Then, at the end of next section, we describe the material in the remaining sections.

2. Notation and fundamental ingredients of the heuristics

Suppose we have n customer segments, with each segment homogeneous (i.e., customers within a segment behave similarly), and m products. R_{ij} denotes the reservation price of customer segment i for product j. Let our decision variables be as follows:

$$\theta_{ij} := \begin{cases} 1 & \text{if customer segment} i \text{buys product } j, \\ 0 & \text{otherwise,} \end{cases}$$

$$\pi_j := \text{price of product } j.$$

Customers strive to maximize their utility which in our case corresponds to *consumer surplus* given by $(R_{ij}-\pi_j)$ for customer i and product j. For a fixed customer segment i, if the consumer surplus is negative for every product then customer segment i would buy nothing. Therefore, customers buy among all products with nonnegative consumer surplus, the one with the largest surplus. We are modeling the problem from the viewpoint of one fixed company with products $1,2,\ldots,m$. To represent the competing companies' products, we assume that the prices of their competing products are known to us. We denote by CS_i (which is part of the data for our optimization problem), the maximum surplus for customer segment i across all competitor products. Then the constraints that model the buying behaviour are:

$$(R_{ij}-\pi_j)\theta_{ij} \ge R_{ik}\theta_{ij} - \pi_k, \quad \forall k \ne j,$$

and

$$(R_{ij}-\pi_j)\theta_{ij} \ge CS_i\theta_{ij}, \quad \forall j,$$

respectively. Note that we may replace R_{ij} by $\max\{R_{ij}-CS_i, 0\}$ for every i,j and assume without loss of generality that $CS_i = 0$ for every i.

Assuming that each customer segment buys at most one type of product, and each customer buys at most one unit of a product, and denoting by N_i the number of customers in segment i, the problem can be expressed as the following nonlinear mixed – integer programming problem:

$$\begin{aligned} \max \ & \sum_{i=1}^{n}\sum_{j=1}^{m} N_i \pi_j \theta_{ij}, \\ \text{s.t.} \ & (R_{ij}-\pi_j)\theta_{ij} \ge R_{ik}\theta_{ij} - \pi_k, \quad \forall j, \forall k \ne j, \forall i, \\ & (R_{ij}-\pi_j)\theta_{ij} \ge 0, \quad \forall j, \forall i, \\ & \sum_{j=1}^{m} \theta_{ij} \le 1, \quad \forall i, \\ & \theta_{ij} \in \{0,1\}, \quad \forall i,j, \\ & \pi_j \ge 0, \quad \forall j. \end{aligned} \tag{1}$$

Note that the second group of constraints (arising from the elimination of CS_i) are very similar to the first group of constraints. Indeed, the second group of constraints can be removed from the formulation by introduction of a dummy product indexed zero, with price set equal to zero. We will see this again shortly when we expose the underlying network structure in the above formulation.

The assumptions stipulated above are not very restrictive in practice as there are ways of relaxing them by modifying the interpretation of the variables; for example, production costs can be incorporated in the objective function without destroying the important mathematical structures of (1). Some of the classical work in the area explicitly included fixed costs. Adding production costs (fixed and/or variable unit costs) to our model does not change the applicability of our algorithms. This is due to the fact that in making assignment decisions (assigning a customer segment to a product), our algorithms evaluate the whole objective function for that assignment/reassignment. As a result, feasible solutions of the problem (1) will be correctly evaluated if we change the objective function of the problem allowing fixed costs. Moreover, the transformations we made in relation to CS_i are indeed still valid in this case.

The assumption about the competitors' prices being stationary and known to us, can also be handled in practice, as long as we are able to reoptimize our prices very quickly (for details, see [19–21]). This provides more motivation for having very fast and effective heuristic algorithms with good re-optimization properties. As we have mentioned in the Introduction, our model is closely related to *envy-free pricing* models. Considering Walrasian Equilibrium Models (going back at least to Walras in 1874, see [22]), equilibrium prices should be chosen so that

(i) customers have no incentive to change their decision;
(ii) the company has no incentive to change its prices.

At an optimal solution of our problem (1), item (i) is satisfied due to the fact that every customer who buys a product, buys a product for which the customer has the maximum possible positive surplus. Item (ii) is satisfied, due to the fact that the total revenue/profit for the company is maximized and no further change to the prices can increase the current revenue/profit.

For a fixed $\theta \in \{0,1\}^{n\times m}$ such that $\sum_{j=1}^{m}\theta_{ij} \le 1$, $\forall i$, the problem (1) becomes an LP problem (on variables π) with a *totally unimodular* (TUM) coefficient matrix. The dual of this LP is equivalent to a shortest path problem, where there are $(m+1)$ nodes (a node for each product and a dummy product). Let 0 denote the dummy product node. Assign arc lengths:

$$r_{kj} := \min_{i \in C_j}\{R_{ij}-R_{ik}\},$$

$$r_{0j} := \min_{i \in C_j}\{R_{ij}\},$$

where $C_j := \{i : \theta_{ij} = 1\}$ and $B := \{j : C_j \ne \emptyset\}$. Note that r_{kj} for $k \in \{1,2,\ldots,m\}$ may be negative and we utilize the usual convention that the minimum over an empty set is $+\infty$. Then, this dual LP