Using MATLAB to Implement a Complementarity-Constrained Pricing Model

Soon Wan Mei Amanda
wanmei.soon@nie.edu.sg
Ang Keng Cheng
kengcheng.ang@nie.edu.sg
Teo Kok Ming
kokming.teo@nie.edu.sg
Mathematics and Mathematics Education Academic Group
National Institute of Education
Nanyang Technological University
Singapore

Abstract: *The demand-price relation in the real market can be approximated by a demand function (DF). A DF can be derived from the first-order conditions of the maximization of a utility function, or estimated using data observed from the markets. In an earlier work, a piecewise nonlinear DF which remain nonnegative at all high prices was proposed, and was called a Complementarity-Constrained Demand Function (CCDF). In this work, we use MATLAB to test a new complementarity-constrained (CC) pricing model. The novel feature of this model is attributed to the CCDF incorporated within it. An algorithm to compare the revenues obtained from a generic pricing model and a CC pricing model was implemented on MATLAB. Results from these computational experiments indicate that the use of a CC pricing model is favourable, for certain ranges of parameters defining the demand function. This demonstrates the important use of computational tools such as MATLAB in pricing modelling studies, and provides justifications for further investigations of the CC pricing model.*

1. Introduction

A demand function (DF) can be used to model the demand-price relation of products in the real market. Some commonly known DFs include linear and Cobb-Douglas DFs. These functions are derived from the first-order conditions of the maximization of utility functions. Though in microeconomics, utility maximization (UM) is a theoretically justified way to generate a DF, it may not enable us to model the real market realistically. In addition, a DF can be generated via UM only when a closed-form solution can be obtained. Thus only a few DFs derived from UM exist.

Data-fitting is an alternative avenue used to determine DFs. That is, we can use some generic functions to fit data that is observed from the real markets. Linear functions are simple and commonly considered in practice. For more examples of DFs considered in the past literature, the reader can refer to [4] and the references therein. However, the DFs are usually defined on a restricted set of prices (we call it Omega), as they turn negative at sufficiently high prices. Thus we are unable to use the functions to account for demand data corresponding to these high prices.

An approach to rectify this problem has been discussed in [5]. In that work, the extension of a DF within Omega to outside Omega is done via the solution of a Complementarity problem, generating a Complementarity-Constrained Demand Function (CCDF) as a result. This CCDF is a piecewise nonlinear DF which remains nonnegative at *all* high prices. The authors proceeded to introduce a new pricing model which incorporated the CCDF.

In this paper, we use MATLAB to test the new complementarity-constrained (CC) pricing model as discussed above. We wrote an algorithm to compare a generic pricing model and a CC pricing model. Through the results of our MATLAB implementations, we observe that a CC pricing model generates higher revenues for certain ranges of parameters defining the demand function. With these numerical results, we proceeded to provide theoretical justifications of the use of the new CC pricing model. Hence, our work demonstrates the important use of MATLAB in the modelling of demand-price relationships of products in a market.

2. Description of a CC Pricing Model

We will briefly discuss the CCDF and the CC pricing model proposed in [5]. For the detailed descriptions of the models and the motivations behind their constructions, the reader can refer to [5].

2.1 Illustration of a CCDF

Consider a demand function $d: R_+^N \to R^N$. Let $\Omega = \{p \in R_+^N \mid d(p) \geq 0\}$. The CCDF is denoted by D, and $D(p) = d(p)$ for all $p \in \Omega$. To define D outside Ω, the following map B defined via a nonlinear complementarity problem (NCP) was introduced in [5].

Definition 2.1 For any $p \in R_+^N$, $B(p)$ is defined as a solution of the $NCP(p)$: find $x\ (= B(p))$ such that

$$0 \leq d(x) \perp p - x \geq 0,$$

where $0 \leq d(x) \perp p - x \geq 0$ represents $d(x) \geq 0, d(x) \cdot (p - x) = 0, p - x \geq 0$.

With the above map, a complete definition of the CCDF follows.

Definition 2.2 The **Complementarity-Constrained Demand Function (CCDF)** $D: R_+^N \to R_+^N$ is defined by

$$D(p) = d\big(B(p)\big), \quad \text{for all } p \in R_+^N,$$

where the map B is as stated in Definition 2.1.

2.2 A simple CC pricing model

We first present a simple generic pricing model involving a demand function defined only on Ω. Consider a single seller offering N products. Let $d(p)$ be a function dependent on the price vector $p \in R_+^N$ of all his products. To maximize his revenue, a seller can solve the following optimization problem:

$$\begin{aligned} \max \quad & p^T d(p) \\ \text{s.t.} \quad & d(p) \geq 0 \\ & p \geq LB, \end{aligned} \tag{2.1}$$

where LB is a given vector of lower bounds on prices to be set by the seller.

If we use a pricing model incorporating the CCDF, non-negative prices outside of Ω are allowed and we have the problem

$$\begin{array}{ll} \max & p^T D(p) \\ \text{s.t.} & p \geq LB. \end{array}$$

Using Definition 2.1, it can be represented as

$$\begin{array}{ll} \max & p^T d(x) \\ \text{s.t.} & 0 \leq d(x) \perp p - x \geq 0 \\ & p \geq LB. \end{array} \tag{2.2}$$

It is clear that if $p = x$ in model (2.2), then models (2.1) and (2.2) are identical. Thus we can always obtain an optimal revenue from model (2.2) that is at least as high as that obtained if model (2.1) was used instead. In this work, our focus is on the cases where model (2.2) generates higher revenues.

3. Comparison of Pricing Models involving a Specific d

In this section, we wish to compare the models (2.1) and (2.2) with a particular form of d incorporated within. For simplicity, we consider a single seller offering two products. As discussed in [6], using past literature including [1], [2] and [3], it is reasonable to consider

$$d_i(p) = c_i p_i^{a_{ii}} p_j^{a_{ij}} - k_i$$

for each product i $(i = 1, 2)$. Here $c_i, k_i,$ a_{ii}, a_{ij} are some given constants, where $c_i, k_i > 0$ are demand parameters, a_{ii} represents the own-price elasticity of demand for product i, and a_{ij} is the cross-price elasticity of demand for product i with respect to product j. It is clear that $a_{ii} < 0$ holds for normal goods and $a_{ij} > 0$ (or < 0) if products i and j are mutually substitutable (or complementary). We will consider only substitutable products in this paper. Note that if $k_i > 0$ and p_j is fixed, the demand d_i goes to 0 before p_i approaches infinity, which seems realistic in the market.

3.1 Solution of a generic pricing problem involving a single seller with two products

Incorporating the function d as discussed above, the generic model (2.1) becomes

$$\begin{array}{ll} \max & p_1\left(c_1 p_1^{a_{11}} p_2^{a_{12}} - k_1\right) + p_2\left(c_2 p_1^{a_{21}} p_2^{a_{22}} - k_2\right) \\ \text{s.t.} & c_1 p_1^{a_{11}} p_2^{a_{12}} - k_1 \geq 0 \\ & c_2 p_1^{a_{21}} p_2^{a_{22}} - k_2 \geq 0 \\ & p_1 \geq LB_1 \\ & p_2 \geq LB_{2.} \end{array} \tag{3.1}$$

To find the optimal solution of problem (3.1), we can use the well-known Karush–Kuhn–Tucker (KKT) conditions. Suppose that $\mu_1, \mu_2, \lambda_1, \lambda_2$ are the lagrange multipliers corresponding to the four respective constraints in (3.1). Then the KKT conditions are:

$$\begin{aligned} -(a_{11}+1)c_1 p_1^{a_{11}} p_2^{a_{12}} + k_1 - a_{21} c_2 p_1^{(a_{21}-1)} p_2^{(a_{22}+1)} - a_{11} k_1 \mu_1 p_1^{(-a_{11}-1)} - a_{21} c_2 \mu_2 p_1^{(a_{21}-1)} - \lambda_1 &= 0 \\ -a_{12} c_1 p_1^{(a_{11}+1)} p_2^{(a_{12}-1)} - (a_{22}+1) c_2 p_1^{a_{21}} p_2^{a_{22}} + k_2 - a_{12} c_1 \mu_1 p_2^{(a_{12}-1)} - a_{22} k_2 \mu_2 p_2^{(-a_{22}-1)} - \lambda_2 &= 0 \\ k_1 p_1^{-a_{11}} - c_1 p_2^{a_{12}} &\leq 0 \end{aligned}$$

$$k_2 p_2^{-a_{22}} - c_2 p_1^{a_{21}} \le 0$$
$$LB_1 - p_1 \le 0$$
$$LB_2 - p_2 \le 0$$
$$\mu_1 \cdot \left(k_1 p_1^{-a_{11}} - c_1 p_2^{a_{12}}\right) = 0$$
$$\mu_2 \cdot \left(k_2 p_2^{-a_{22}} - c_2 p_1^{a_{21}}\right) = 0$$
$$\lambda_1 \cdot (LB_1 - p_1) = 0$$
$$\lambda_2 \cdot (LB_2 - p_2) = 0$$
$$\mu_1, \mu_2, \lambda_1, \lambda_2 \ge 0$$

There are sixteen cases involved in the solution of the KKT conditions above. Some examples are $\mu_1, \mu_2, \lambda_1, \lambda_2 > 0$ and $\mu_1, \mu_2, \lambda_1 > 0, \lambda_2 = 0$. The basic idea is to compare the solutions of all the cases and identify the price vector that gives the maximum objective value. A more detailed discussion of the solution procedure of problem (3.1) can be found in [6]. We can use MATLAB to program an algorithm to solve the sixteen cases and find the optimal revenue, for given values of the parameters and lower bound constraints.

3.2 Solution of a CC pricing problem involving a single seller with two products

If model (2.2) is considered , use of the same function d as above leads to the problem

$$\begin{array}{ll} \max & p_1\left(c_1 x_1^{a_{11}} x_2^{a_{12}} - k_1\right) + p_2\left(c_2 x_1^{a_{21}} x_2^{a_{22}} - k_2\right) \\ \text{s.t.} & 0 \le c_1 x_1^{a_{11}} x_2^{a_{12}} - k_1 \perp p_1 - x_1 \ge 0 \\ & 0 \le c_2 x_1^{a_{21}} x_2^{a_{22}} - k_2 \perp p_2 - x_2 \ge 0 \\ & p_1 \ge LB_1 \\ & p_2 \ge LB_{2.} \end{array} \tag{3.2}$$

The following four cases are important in the consideration of model (3.2):

(I) $d_1(x) = d_2(x) = 0,\ p_1 - x_1 \ge 0,\ p_2 - x_2 \ge 0.$
(II) $p_1 - x_1 = 0,\ p_2 - x_2 = 0,\ d_1(x) \ge 0,\ d_2(x) \ge 0.$
(III) $d_1(x) = 0,\ p_2 - x_2 = 0,\ p_1 - x_1 \ge 0,\ d_2(x) \ge 0.$
(IV) $d_2(x) = 0,\ p_1 - x_1 = 0,\ d_1(x) \ge 0,\ p_2 - x_2 \ge 0.$

It is easy to see that, to compare the optimal revenues obtained from models (3.1) and (3.2), we just need to find the objective function values of problem (3.2) under Cases (III) and (IV), and then compare them with that of model (3.1). As solving each of these two cases can be reduced to solving a constrained single-variable maximization problem (see [6] for details), the solution method is not difficult to program using MATLAB. The reader can refer to the Appendix for the program.

4. Numerical Observations

Using MATLAB, we programmed an algorithm to implement the solution procedures discussed previously to compare the two pricing models (3.1) and (3.2). For simplicity of exposition, we consider the fixed parameters $c_1 = c_2 = k_1 = k_2 = a_{12} = a_{21} = 1$ throughout this section. These numbers were chosen for convenience of our qualitative study of the models. Note that $a_{12} = a_{21}$

implies that the two products are equally substitutable for each other, which can happen in the real market. We are in the process of calibrating our model to fit a real situation using data on ebay. The results will be reported in a later paper.

Due to the fixed parameters defining the functions d_1 and d_2, it is easy to verify that $d_1(p) = d_2(p) = 0$ at $p = (1,1)$ [implicit upper bounds on prices]. Thus we consider $LB_i < 1$ $[i = 1,2]$. We now present the graphs illustrating some of the numerical experiments that we undertook. There are three sets of cases with three cases in each set, namely, A1- A3, B1-B3 and C1-C3. For example, in all the Cases A1-A3, LB_1 is fixed at 0.5, a_{22} ranges from -10 to -3 and LB_2 ranges from 0.5 to 0.95. Thus only the parameter a_{11} is different across the 3 cases. However, the graphs of these 3 cases are the same, hence we only present one representative graph here. The descriptions of all the different cases are provided within the captions of the figures. Note that if the maximum revenue is obtained "outside Ω", then it means that model (3.2) is better.

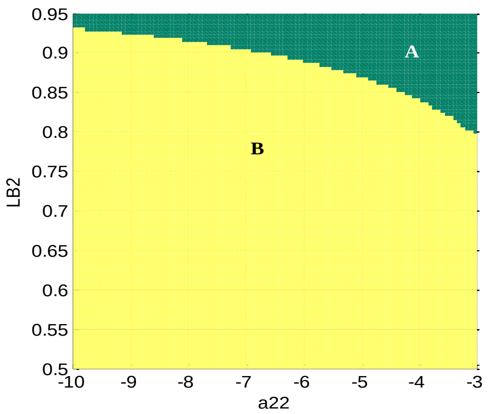

Figure 4.1 Cases A1-A3

Regions A and B are the sets of parameters where maximum revenues are obtained outside and inside Ω respectively, with $LB_1 = 0.5, a_{11} = -6$ (Case A1), $a_{11} = -4$ (Case A2) and $a_{11} = -2$ (Case A3).

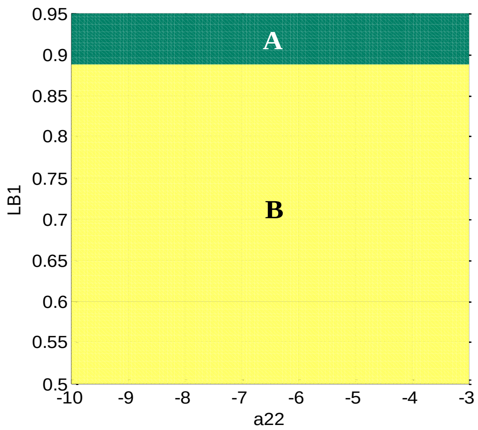

Figure 4.2 Case B1

Regions A and B are the sets of parameters where maximum revenues are obtained outside and inside Ω respectively, with $a_{11} = -6$ and $LB_2 = 0.5$.

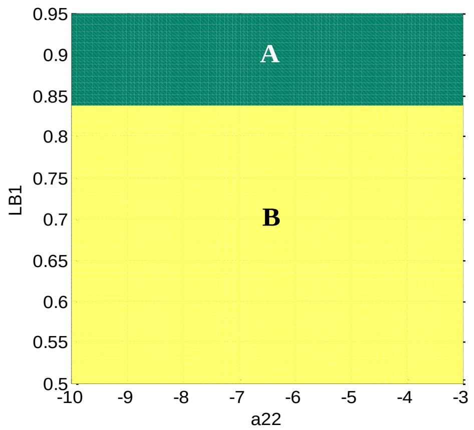

Figure 4.3 Case B2

Regions A and B are the sets of parameters where maximum revenues are obtained outside and inside Ω respectively, with $a_{11} = -4$ and $LB_2 = 0.5$.

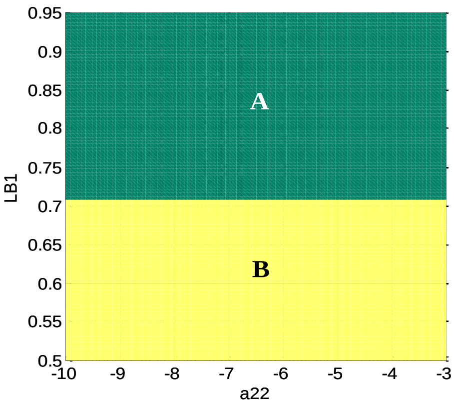

Figure 4.4 - Case B3
Regions A and B are the sets of parameters where maximum revenues are obtained outside and inside Ω respectively, with $a_{11} = -2$ and $LB_2 = 0.5$.

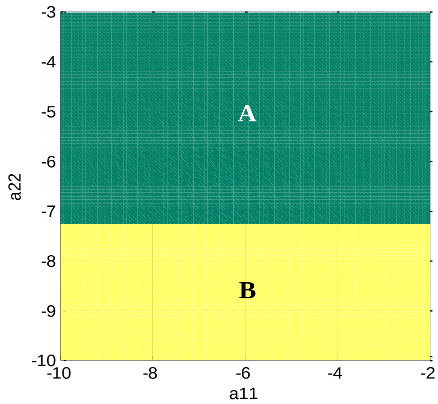

Figure 4.5 - Case C1
Regions A and B are the sets of parameters where maximum revenues are obtained outside and inside Ω respectively, with $LB_1 = 0.2$ and $LB_2 = 0.8$.

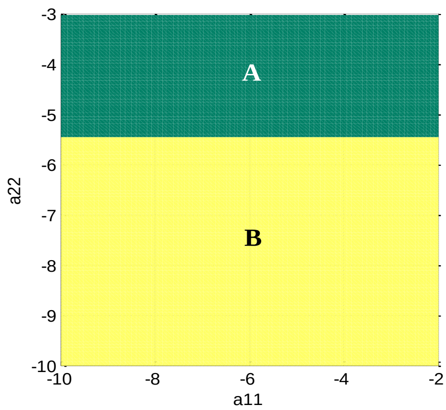

Figure 4.6 - Case C2
Regions A and B are the sets of parameters where maximum revenues are obtained outside and inside Ω respectively, with $LB_1 = 0.3$ and $LB_2 = 0.8$.

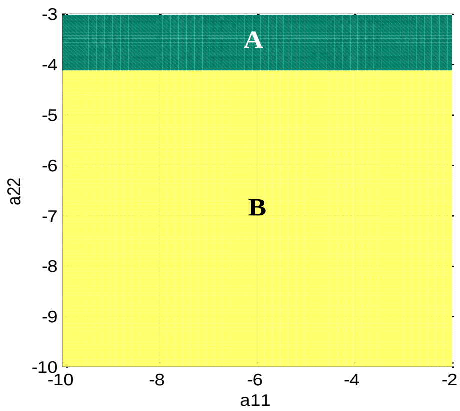

Figure 4.7 - Case C3
Regions A and B are the sets of parameters where maximum revenues are obtained outside and inside Ω respectively, with $LB_1 = 0.4$ and $LB_2 = 0.8$.

From studying the graphs above and the output values generated from our customized MATLAB programs, we were able to obtain the threshold values which, when exceeded, cause the transitions from regions B to regions A.

For each of the cases A1-A3, the value of LB_1 is fixed. Since the graphs are identical though a_{11} varies across the three cases, it is clear that the threshold values do not depend on a_{11}. In addition, from Figure 4.1, we observe that the threshold value of LB_2 decreases as a_{22} increases. We can illustrate the threshold values using the figures below.

Let x^* be the vector where $d_2(x^*)= 0$ and $x_1^* = LB_1$ [see Figure 4.8]. Then if $LB_2 > x_2^*$, it is possible to set $p^* = (LB_1, LB_2)$ and x^* as discussed, under Case (IV) of model (3.2). Under model (3.1), the feasible region of prices is restricted by Ω and the lower bound constraints [denoted by R

in Figure 4.8]. However, when $LB_2 < x_2^*$, there is no advantage in considering model (3.2) as there will no longer be a situation where we can set $p_i = LB_i$ and $x_i < LB_i$ $[i = 1, 2]$. Thus the threshold value of LB_2 is x_2^*. From the expression $d_2(LB_1, x_2^*) = 0$, we can obtain $x_2^* = (LB_1)^{-1/a_{22}}$. It is clear that this expression does not contain the a_{11} term. The correctness of this expression can be verified using the output values of our MATLAB programs. From Figure 4.9, we can see that this threshold value of LB_2 decreases as a_{22} increases [i.e., $\bar{x} < \bar{\bar{x}}$].

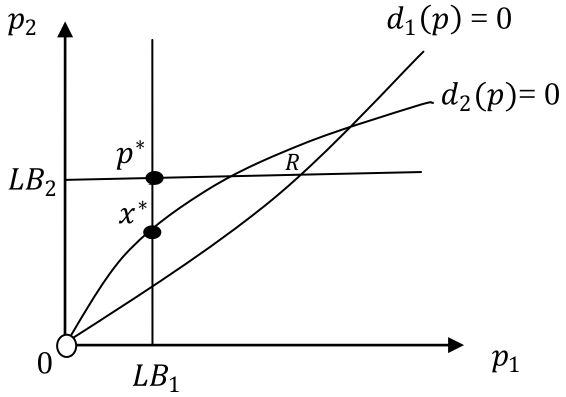

Figure 4.8 Optimal p^* outside Ω and x^* in Ω

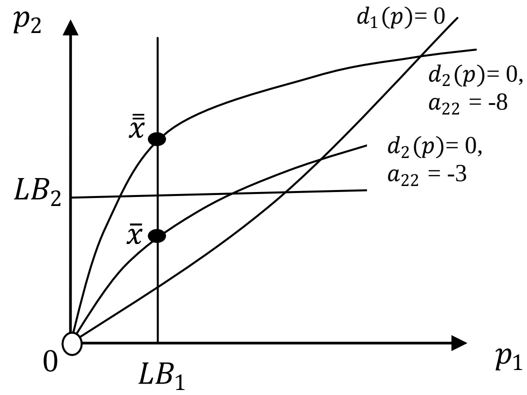

Figure 4.9 Dependence of threshold values on a_{22}

For Cases B1-B3, LB_2 is fixed and a_{11} varies across the three cases. It is clear that the threshold value of LB_1 for each case is independent of a_{22}. We also observe that as a_{11} increases, the threshold value decreases. See Figures 4.2-4.4.

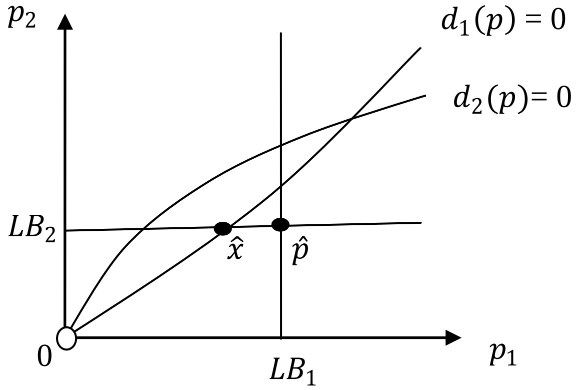

Figure 4.10 Optimal $\hat{p}$ outside Ω and $\hat{x}$ on boundary of Ω

Similar to the earlier discussion, suppose we now denote $\hat{x}$ as the vector where $d_1(\hat{x}) = 0$ and $\hat{x}_2 = LB_2$ [refer to Figure 4.10]. Once $LB_1 > \hat{x}_1$, it is possible to set $\hat{p} = (LB_1, LB_2)$ and $\hat{x}$ as defined, under Case (III) of model (3.2). Hence the threshold value of LB_1 is $\hat{x}_1$. From the expression $d_1(\hat{x}_1, LB_2) = 0$, we obtain $\hat{x}_1 = (LB_2)^{-1/a_{11}}$. Since $LB_2 < 1$ and $a_{11} < 0$, we can easily see why the threshold value decreases as a_{11} increases.

In the last set of cases, i.e., C1-C3, we fix LB_2 and vary LB_1 across the cases. According to Figures 4.5-4.7, the threshold values are independent of a_{11} and increase as LB_1 increases. Note that both LB_1 and LB_2 are fixed within each case. We can visualize using Figure 4.9, that once a_{22} increases beyond a certain threshold value [say $\bar{a}_{22}$], we can set $p^* = (LB_1, LB_2)$, $x_1^* = LB_1$ and

$x_2^* = (LB_1)^{-1/a_{22}}$ [see the discussion on Cases A1-A3]. Higher revenues can then be obtained using model (3.2). It is easy to see that $\bar{a}_{22}$ occurs at $d_2(LB_1, LB_2) = 0$. That is, $\bar{a}_{22} = \frac{-\ln LB_1}{\ln LB_2}$. As before, we can verify this expression using our MATLAB output values. In addition, with $LB_1, LB_2 < 1$, it is clear that $\bar{a}_{22}$ increases with LB_1.

We are able to prove the mathematical expressions for the threshold values as discussed above, with the help of our MATLAB programs. The main idea is to show that higher revenues can be obtained using model (3.2) for parameters within the region A. As the methods of proofs for all the three types of cases are similar, we will only discuss the proof for Cases A1-A3. From Section 3.1, we saw that sixteen cases constitute the solutions of the KKT conditions of problem (3.1). Then in Section 3.2, we showed that we need to solve problem (3.2) under Cases (III) and (IV). However, from our MATLAB results, we observed that only three of the eighteen cases yielded non-trivial solutions. Using algebraic manipulations and reasoning specific to Cases A1-A3, we were able to show that, under certain trivial conditions, fifteen of the eighteen cases were inadmissible and can be ignored. We then found the solutions of the remaining three non-trivial cases and proved that the revenue obtained under Case (IV) of model (3.2) is the highest [for any given set of parameters in region A]. This showed that for certain parameters defining the demand function and lower bound constraints, the CC pricing model is indeed favourable. All technical details of our proofs will be reported in a later work.

5. Conclusion and further work

The use of MATLAB was instrumental in comparing a generic pricing model and our complementarity-constrained pricing model. We have shown through the numerical examples that, for a simple single-seller-two-product problem, we should not ignore prices outside Ω as they can lead to better revenues. Our MATLAB results were also useful in pointing the direction of our mathematical proofs. Henceforth, we can model the function d using real data and conduct simulation studies of the demand-price relationships of certain specific products using MATLAB. An investigation of the two pricing models incorporating a realistic d will then follow.

In addition, we can incorporate our project into the teaching of an Honours or Masters Operations Research (OR) course. Students can first be guided to obtain real data (for example, from ebay) to estimate the parameters defining the function d. They can then input these parameters into our given program to obtain the output of optimal prices and revenues, and analyse and compare pricing models. In this way, they can gain a deeper understanding of pricing models and learn a real application of OR techniques, without the need to write complicated programs.

Acknowledgements This research is supported partly by NIE Academic Research Fund Project Reference No: RI 1/08 SWM. The authors would like to thank Mr Wee Wen Shih, their Research Assistant, for his help in this project.

References

[1] Chevalier, J., and Goolsbee, A. (2003). *Measuring Prices and Price Competition Online: Amazon.com and Barnes and Noble.com.* Quantitative Marketing and Economics, 1, 203-222.

[2] Ghose, A., Smith, M. D., and Telang, R. (2006). *Internet Exchanges for Used Books: An Empirical Analysis of Product Cannibalization and Welfare Impact*. Information Systems Research, 17 (1), 3-19.
[3] Ghose, A., and Sundararajan, A. (2006). *Evaluating Pricing Strategy Using e-Commerce Data: Evidence and Estimation Challenges*. Statistical Science, 21(2), 131-142.
[4] Soon, W. M. (2009). A review of multi-product pricing models. Submitted to Applied Mathematics and Computation.
[5] Soon, W. M., Zhao, G. Y., and Zhang, J. P. (2009). *Complementarity demand functions and pricing models for multi-product markets.* European Journal of Applied Mathematics, 20(5), 399-430.
[6] Soon, W. M., Ang, K. C., and Teo, K. M. (2009). *Numerical Implementations of a Pricing Model with Complementarity Constraints.* In Yahya Abu Hasan et al. (Ed.) 5th Asian Mathematical Conference Proceedings (pp. 205-211). Kuala Lumpur: School of Mathematical Sciences, Universiti Sains Malaysia.

Appendix (Program for Section 3.2)

```
%Parameters given
a11 = -3.5;
a12 = 3.24;
a21 = 0.23;
a22 = -0.45;
c1 = 1;
k1 = 1;
c2 = 1;
k2 = 1;
L1 = 0.5;
L2 = 0.8;

%To simplify notations for the program
ck1 = c1/k1;
ck2 = c2/k2;
kc1 = k1/c1;
kc2 = k2/c2;
d = a11*a22 - a12*a21;

% ------------------------------------------------------------------------------------
%To solve Case (III): p_2 = x_2, d_1(x_1,x_2) = 0

UB2 = (ck2*(kc1^(a21/a11)))^(-a11/d);   %find upper bound on x_2
x2 = (kc2*(ck1^(a21/a11))*(a11/(a11+d)))^(a11/d);   %find critical point
fL2 = c2*(kc1^(a21/a11))*L2^(1+(d/a11)) - k2*L2;   %find function values at endpoints
fUB2 = c2*(kc1^(a21/a11))*UB2^(1+(d/a11)) - k2*UB2;

if UB2 < L2    %this case is infeasible
  Rev2 = -1;
  p1 = -1;
  p2 = -1;
else
  if L2 < x2 & x2 < UB2 & isreal(x2) == 1
```

```
      fx2 = c2*(kc1^(a21/a11))*x2^(1+(d/a11)) - k2*x2; %find function value at critical pt
      [Rev3,I] = max([fL2 fUB2 fx2]);
      if I == 1
        p2 = L2;
      elseif I == 2
        p2 = UB2;
      else
        p2 = x2;
      end
      p1 = max([L1 (kc1*(p2^(-a12)))^(1/a11)]);
      else
         [Rev3,I] = max([fL2 fUB2]);
      if I == 1
        p2 = L2;
      else I == 2
        p2 = UB2;
      end
      p1 = max([L1 (kc1*(p2^(-a12)))^(1/a11)]);
    end
  end
fprintf('CC Case (III): p1 = %g, p2 = %g, revenue = %g \n',p1,p2,Rev3);

% ------------------------------------------------------------------------------------
%To solve Case (IV): p_1 = x_1, d_2(x_1,x_2) = 0

UB1 = (ck1*(kc2^(a12/a22)))^(-a22/d);   %find upper bound on x_1
x1 = (kc1*(ck2^(a12/a22))*(a22/(a22+d)))^(a22/d);   %find critical point
fL1 = c1*(kc2^(a12/a22))*L1^(1+(d/a22)) - k1*L1;   %find function values at endpoints
fUB1 = c1*(kc2^(a12/a22))*UB1^(1+(d/a22)) - k1*UB1;

if L1 < x1 & x1 < UB1 & isreal(x1) == 1
  fx1 = c1*(kc2^(a12/a22))*x1^(1+(d/a22)) - k1*x1;
  [Rev4,I] = max([fL1 fUB1 fx1]);
  if I == 1
    p1 = L1;
  elseif I == 2
    p1 = UB1;
  else
    p1 = x1;
  end
  p2 = max([L2 (kc2*(p1^(-a21)))^(1/a22)]);
else
  [Rev4,I] = max([fL1 fUB1]);
  if I == 1
    p1 = L1;
  else I == 2
    p1 = UB1;
  end
  p2 = max([L2 (kc2*(p1^(-a21)))^(1/a22)]);
end
fprintf('CC Case (IV): p1 = %g, p2 = %g, revenue = %g \n',p1,p2,Rev4);
```