Department of Mathematics and Statistics

# Joint Pricing and Production Planning of Multiple Products

Elham Mardaneh

This thesis is presented for the degree of
Doctor of Philosophy
of
Curtin University

September 2010

# CERTIFICATION

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made. This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Elham Mardaneh

September 2010

# Abstract

Many industries are beginning to use innovative pricing techniques to improve inventory control, capacity utilisation, and ultimately the profit of the firm. In manufacturing, the coordination of pricing and production decisions offers significant opportunities to improve supply chain performance by better matching supply and demand. This integration of pricing, production and distribution decisions in retail or manufacturing environments is still in its early stages in many companies. Importantly it has the potential to radically improve supply chain efficiencies in much the same way as revenue management has changed the management of the airline, hotel and car rental industries. These developments raise the need and interest of having models that integrate production decisions, inventory control and pricing strategies.

In this thesis, we focus on joint pricing and production planning, where prices and production values are determined in coordination over a multiperiod horizon with non-perishable inventory. We specifically look at multiproduct systems with either constant or dynamic pricing. The fundamental problem is: when the capacity limitations and other parameters like production, holding, and backordering costs are given, what the optimal values are for production quantities, and inventory and backorder levels for each item as well as a price at which the firm commits to sell the products over the total planning horizon. Our aim is to develop models and solution strategies that are practical to implement for real sized problems.

We initially formulate the problem of time-varying pricing and production planning of multiple products over a multiperiod horizon as a nonlinear programming problem. When backorders are not allowed, we show that if the demand/price function is linear, as a special case of the without backorders model, the problem becomes a Quadratic Programming problem which has only linear constraints. Existing solution methods for Quadratic Programming problem are discussed. We then present the case of allowed backorders. This assumption

makes the problem more difficult to handle, because the constraint set changes to a non-convex set. We modify the nonlinear constraints to obtain an alternative formulation with a convex set of constraints. By this modification the problem becomes a Mixed Integer Nonlinear Programming problem over a linear set of constraints. The integer variables are all binary variables. The limitation of obtaining the optimal solution of the developed models is discussed. We describe our strategy to overcome the computational difficulties to solve the models.

We tackle the main nonlinear problem with backorders through solving an easier case when prices are constant. This resulting model involves a nonlinear objective function and some nonlinear constraints. Our strategy to reduce the level of difficulty is to utilise a method that solves the relaxed problem which considers only linear constraints. However, our method keeps track of the feasibility with respect to the nonlinear constraints in the original problem. The developed model which is a combination of Linear Programming (LP) and Nonlinear Programming (NLP) is solved iteratively. The solution strategy for the constant pricing case constructs a tree search in breadth-first manner. The detailed algorithm is presented. This algorithm is practical to implement, as we demonstrate through a small but practical size numerical example.

The algorithm for the constant pricing case is extended to the more general problem. More specifically, we reformulate the time-variant problem in which there are multi blocks of constant pricing problems. The developed model is a combination of Linear Programming (LP) and linearly constrained Nonlinear Programming (NLP) which is solved iteratively. Iterations consist of two main stages: finding the value of LP's objective function for a known basis, solving a very smaller size NLP problem. The detailed algorithm is presented and a practical size numerical example is used to implement the algorithm. The significance of this algorithm is that it can be applied to large scale problems which are not easily solved with the existing commercial packages.

We include the uncertainty of the demand/price function in this thesis by considering a set of scenarios. The purpose of this effort is to develop a robust optimisation (RO) model to determine the optimal production planning and constant pricing of a manufacturing system with multiple products over a multiple period horizon to maximize the total profit. We illustrate our model and its solution with two practical size examples. The importance of this model and its solution strategy is the novel use of robust optimisation in the discrete case of joint pricing and production planning.

**To my family**

**for their constant love and support …**

# AKNOWLEDGEMENTS

First, I would like to thank my thesis advisor, Professor Louis Caccetta, for his guidance throughout my graduate process. It would not have been possible without his unfailing energy, his wisdom, and his understanding. I appreciate his insight on all manners of things, both research and non-research related. Truly, I am grateful for all he has done. I appreciate the support of Professor Yong Hong Wu for his role as graduate chair.

I would like to thank my husband, Amir Miri, who always has confidence in me and always stays with me to overcome whatever difficulty I face in my daily life during my Ph.D. study.

Much appreciation goes to Curtin University for their financial support through the provision of a Curtin Post-Graduate Award with stipend scholarship.

Finally my thanks go out to administration staff and graduate students in this department and my friends. The ones who have gone before me provided inspiration as well as guidance. The ones with me shared the struggles and the joys

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1

# Introduction

In recent years, retail and manufacturing companies look for innovative pricing strategies to improve their operations and optimise their outputs. Various tools have been developed and utilised for this purpose including: dynamic pricing over time; target pricing to different classes of customers and pricing to understand customer demand. The advantages of these tools include a potential increase in profit and having less variability in demand or production, which cannot be ignored.

A significant revolution in retail and manufacturing industries is using Internet or the Direct-to-Customer model to dynamically change the price of products. Streitfeld (2000) reports that big companies such as Dell Computers and Amazon.com use the Direct-to-Customer model to change prices quickly and easily based on parameters such as demand variation, inventory levels, or production schedules. The model also enables manufacturers to collect demand data more simply and precisely.

Utilising a Direct-to-Customer model over the Internet brings an easy implementation of price changes with a very low cost involved, because there is no need to produce catalogues or price stickers. Also, the Internet provides a novel source of information about customer demand and priorities.

Traditionally, many firms and researchers focus on pricing alone as a tool to improve profits. However for manufacturing industries, the coordination of price decisions with other aspects of the supply chain such as production and distribution is not only useful, but is essential. The coordination of these decisions means an approach that optimises the system rather than individual elements, improving both the efficiency of the supply chain and of the firm. This integration of pricing, production and distribution decisions in retail or manufacturing environments is still in its early stages in many companies. Importantly it has the potential to radically improve supply chain efficiencies in much the same way as revenue management has changed the management of the airline, hotel and car rental industries. These developments raise the need and interest of having models that integrate production decisions, inventory control and pricing strategies. Thus in this study, we discuss several models that combine pricing with other aspects of supply chain, particularly those related to production planning or inventory control. We characterize the optimal inventory/production and pricing strategies and develop computational algorithms for determining the optimal policies.

This Chapter is organised as follows. Section 1.1 gives an introduction to the joint pricing and production planning problems, in particular for multiple products over multiperiod horizon systems. Essential features and the main objective of this project are detailed. In Section 1.2 we provide the current literature that is closely related to this project. Section 1.3 provides a brief outline of this thesis.

## 1.1. The problem

A fundamental problem in joint pricing and production planning is finding efficient inventory controls/production plans and pricing strategies for satisfying the endogenous demands over the planning horizon. Usually the plan and chosen price should satisfy some constraints. The efficiency of the joint pricing and production planning can be measured in terms of profit. The constraints of above problem can consist of production capacity, the need to produce a specific amount in some stages of the horizon and how to deal with the excess demand.

To define the problem more clearly, we find it useful to describe briefly the literature of joint pricing and production planning. Simchi-Levi et al. (2004) group papers according to a number of characteristics of the problem or assumptions made by the researchers. These characteristics are as follows:

- **Length of Horizon**

For the finite horizon, a number of publications consider a single period problem, similar to a newsvendor setting. Occasionally, researchers assume a two period problem. Other possibilities include a multiple period for the both finite and infinite horizon.

- **Prices**

For the multiple period problems, price may remain fixed or constant over the time horizon, even if the demand is non-stationary over time. Another classification allows for price to dynamically change over time (as a function of demand, inventory, or other parameters of the problem). This is particularly important for firms that sell products through electronic channels.

- **Demand Type**

The first distinction about demand is whether it is deterministic, with a known function according to parameters like price, or stochastic or random. Generally in the case of uncertainty, it is assumed that there is some known portion that is based on price (e.g., linear demand curve), with an additional stochastic element.

- **Sales**

In the case of stochastic demand, or when production capacity limits exist, researchers make assumptions about how to treat excess demand. The primary assumptions are that demand is either backordered or lost. In some cases neither assumption is needed, for example under deterministic demand with no limits or when price can be set to match demand exactly.

- **Production Set-up Cost**

For some manufacturing problems the addition of a fixed production set-up cost may be appropriate. In general, the addition of this fixed charge complicates the structure of the objective function and makes the problems more difficult to solve (also noted in Eliashberg and Steinberg (1991) as a distinction between convex or non-convex cost functions).

- **Capacity Limits**

For manufacturing problems, researchers sometimes include the fact that production may be limited by the capacity of the system. Although the default is that most papers do not consider capacity limits.

- **Products**

Most publications consider a single product or multiple products that do not share resources, in which case the problems are separable into the single product case [Gilbert (1999), Bernstein & Federgruen (2003), Biller et al. (2005), Elmaghraby et al (2008), Neale (2009)]. A few researchers consider multiple products that share common production resources or components, or share demand from customers [Gilbert (2000), Kachani and Perakis (2002)].

Now, we define the major problem we address in this thesis. We focus on joint optimisation of pricing and production planning of a multiple product manufacturing system over a multiperiod horizon. In this system, the price is assumed to be either constant or time-varying. The case of constant pricing is a base attempt to find the optimal solution of the time-varying problem. Demand is deterministic and a function of price, although we consider uncertainty in the relationship of demand and price in Chapter 5. For the deterministic demand, both cases of with and without backorders are discussed. The excess demand is considered only for the deterministic case. Production setup cost is negligible. The capacity is shared amongst multiple products.

Given the capacity limitations and other parameters like production, holding or backordering costs, our objective is to decide upon production quantities, inventory and backorder levels for each item as well as a price at which the firm commits to sell the products over the total planning horizon.

In the next Section the literature of joint pricing and production planning which is closely related to our project is discussed.

## 1.2. Literature Review

In this Section, we first review the fixed pricing models that are mostly limited to a single product problem. We then review the literature on joint optimization of pricing and production planning for multiple products.

Although much of the research in pricing and inventory control has centred on dynamic prices [Rajan et al. (1992), Zhao and Zheng (2000), Vizard et al. (2001), Petruzzi and Dada (2002), Netessine (2006)], some [Kunreuther and Schrage (1973), Gilbert (1999 and 2000), Chan et al. (2006)] has also considered the problem of choosing a fixed or constant price over the lifetime of a product.

The earliest known example of a problem that integrates a fixed price decision with inventory policies is that of Kunreuther and Schrage (1973). They consider the case of a single product with demand that is deterministic. Demand is a linear function of price, and varying over a season, and they include production set-up costs. Their model does not have lost sales or backlogging, since demand is exactly that predicted by price and time, and there are no production capacity limits. The objective is to determine price, production schedules, and production quantities so as to maximize profit. The authors provide a ”hill-climbing” algorithm that provides upper and lower bounds on the price decision. “Hill climbing” is defined by Russell and Norvig (2003) as a mathematical optimization technique which belongs to the family of local search methods. It can be used to solve problems that have many solutions, some of which are better than others. It starts with a random (potentially poor) solution, and iteratively makes small changes to the solution, each time improving it a little. When the algorithm cannot see any improvement, it terminates. Ideally, at that point the current solution is close to optimal, but it is not guaranteed that hill climbing will ever come close to the optimal solution.

The variant of the problem in which a single price must be chosen for the entire horizon was studied by Gilbert (1999). Using a slightly less general model of demand and imposing the restriction that setup and holding costs be time-invariant, he shows that the total cost of production and inventory is a piecewise linear function of price. He exploits this property to develop a solution approach that guarantees optimality for the problem, employing a Wagner-Whitin time approach for determining production periods. The Wagner-Whitin algorithm is a dynamic programming lot sizing model that evaluates multiple alternatives that consider period demand and production, holding, and setup costs to produce an optimal lot size that varies for each period as required. Gilbert (2000) extends this research to the problem of determining a single price for each of a number of goods when the goods share production capacity. He again assumes that revenue is concave, but does not consider production set-up cost in the multiple product model. Demand is a function of time and product characteristics, and is multiplicative with seasonality, i.e., $D_{jt} = \beta_{jt}D_j$, where $D_j$ represents the demand intensity, $\beta_{jt}$ is the seasonality factor of item $j$ in period $t$ and $D_{jt}$ is the induced demand of item $j$ in period $t$. By formulating a deterministic optimization problem and using the dual, Gilbert develops an iterative algorithm that solves the problem to optimality. By applying the procedure to a numerical example, Gilbert also demonstrates that a firm that is pricing multiple products may want to be more aggressive in pricing products that have high demand early in the season. Further sensitivity shows that if a product has greater seasonality than another product, the price may be higher.

Although most of the work on constant pricing assumes deterministic demand, fixed pricing under stochastic demand is a special case of the Delayed Production problem in Chan et al (2006). In this research, the authors consider a general stochastic demand function over multiple periods, where production capacity is limited but set-up costs are not incurred. Excess demand is lost, and sales are discretionary, i.e., inventory may be set aside to satisfy future demand even at the expense of lost sales in the current period. The authors develop a dynamic programming model that solves the problem to optimality for discrete possibilities of fixed price. They describe policies and heuristics for the strategies based on deterministic approximations and analyse their performances. In a numerical study, making the production and price decisions under stochasticity is more important when there is limited deterministic seasonality but high levels of uncertainty about the actual demand realization.

Now we turn specifically to the papers which consider multiple products. The product line design problem (see Yano and Dabson (1998) for a review) is concerned with the selection of a mix of products to offer in the market. As Morgan et al. (2001) point out, this problem has typically been considered from a marketing perspective, while the operational aspects of product line decisions have been largely ignored. Morgan et al. (2001) consider individual product costs and relevant cost interactions among products in their product line design model, but the prices of products are given as inputs.

Bassok et al. (1999) consider a model with $N$ products and $N$ demand classes with full downward substitution, i.e., excess demand for class $i$ can be satisfied using product $j$ for $i \geq j$. They show that a greedy allocation policy is optimal. However, there are no pricing decisions in their model. Meyer (1976) considers a multiproduct monopoly pricing model under risk. The firm must make decisions for prices, productions, and capacities before actual demand is known. However, his model does not consider explicitly the inventory related costs.

There are other papers which consider a pricing and inventory model where the demand is a function of the pricing decision. Most of this work is extendable to multiple products that do not share a common resource.

A single period paper that considers the pricing of multiple products with substitution is Birge et al (1998). The authors also set the capacity levels for production. They consider a single-period model in which a firm produces two products with price-dependent demands. The firm has the ability to make pricing or capacity decisions for one or both of its products. By assuming the demands to be uniformly distributed, they are able to show that the pricing and capacity decisions are affected greatly by the system parameters that the decision makers can control. They consider two different cases. In the first case, the capacity is fixed for both products, but the firm can set prices. In the second case, each product is managed by a product manager trying to maximize individual product profits rather than overall firm profits and analyse how optimal price and capacity decisions are affected.

A few papers with multiple time periods have also considered multiple products. Gallego and Van Ryzin (1997) focus on a multi-market problem, with multiple products sharing common resources. There is a finite horizon over which the firm can sell its products. They model demand as a stochastic point process function of time and the prices of all products. Revenue is assumed to be concave. Sales are neither backordered nor lost, as price is set to infinity when inventory is zero ("null- price condition: the price at which demand falls to zero"). Gallego and Van Ryzin (1997) formulate a deterministic problem, which they show gives a bound on the expected revenue. This problem also motivates the creation of a make-to-stock (MTS) and a make-to-order (MTO) heuristic. The MTS heuristic requires that all products be preassembled, and the price path is determined from the deterministic solution. The MTO heuristic also uses the prices from the deterministic solution but produces and sells products as they are requested. An order is rejected if the components are not available to assemble it. The authors show that each of these heuristics is asymptotically optimal as the expected sales increases.

Smith et al. (1998), develop a model to plan promotions and advertising, and use scenarios to represent market conditions. The decisions of the firm are the promotion price and the advertisement size or cost, where the advertisement is limited by a budget. The deterministic demand depends on the price as well as the advertisement type, and demand scenarios occur with some probability specified by the user. With additional constraints such as a limit on the number of markdowns, the authors develop an optimization problem that maximizes expected profit over multiple products.

Biller et al. (2005) analyse a pricing and production problem where (in extensions), multiple products may share limited production capacity. When the demand for products is independent and revenue curves are concave, the authors show that an application of the greedy algorithm provides the optimal pricing and production decisions. Excess demand is lost. Black (2005) defines the greedy algorithm as any algorithm that follows the problem solving metaheuristic of making the locally optimal choice at each stage with the hope of finding the global optimum.

Kachani and Perakis (2002) study a pricing and inventory problem with multiple products sharing production resources, and they apply fluid dynamic methodology to make their pricing, production, and inventory decisions. In their case, they consider the sojourn or delay time of a product in inventory, where the delay is a deterministic function of initial inventory and price (including competitor's prices). For the continuous time formulation, they establish when the general model has a solution, and for the discretised case they provide an algorithm for computing pricing policies.

Recently, Zhu and Thonemann (2009) studied a pricing and inventory control problem for a retailer who sells two products. The demand of each product depends on the prices of both products, that is, the cross price effects exist between two products. They show that the base-stock list price policy, which is optimal for the single product problem, is no longer optimal for their problem. They derive the optimal pricing and inventory-control policy and show the retailer can greatly improve profits if it manages the two products jointly and consider the cross price effects. Specifically, they prove that the optimal pricing policy depends on the starting inventory levels of both products and the optimal expected profit is sub modular in the inventory levels.

Karakul and Chan (2008) consider a company that produces a well-established product and wants to introduce a new product. The existing product is priced at its optimum equilibrium value and has a stable pool of major customers. The new product targets for a more demanding market segment and has all the functionality of the existing product. As a result, the company has the option to offer the new product as a substitute at a cut price in case the existing product runs out. Both products are seasonal and have fairly long production lead-times. The company faces the single period problem of pricing the new product and having optimal quantities of both products on hand till replenishment is possible. Demand of the existing product during the period is represented by a discrete distribution generated by the pool of major customers. However, demand of the new product has to be estimated and is represented by a set of price dependent continuous distributions. The objective is to find the optimal price for the new product and inventory level for both products so as to maximize the single period expected profit.

The authors show that the problem can be transformed to a finite number of single variable optimization problems. Moreover, for some general new product demand distributions, the single variable functions to be optimized have only two possible roots each. These demand distributions include Normal, Log-Normal, Uniform, Exponential, Gamma, etc. They also show that besides the expected profit, both the price and production quantity of new products are higher when it is offered as a substitute.

Fluid dynamic models are also used by Adida and Perakis (2007) to study a make-to-stock manufacturing system with deterministic demand. They introduce and study an algorithm that computes the optimal production and pricing policy as a function of the time on a finite time horizon, and discuss some insights. Their results illustrate the role of capacity and the effects of the dynamic nature of demand in the model.

In Section 1.3 we give a brief review and outline of the thesis.

## 1.3. Review and Outline of Thesis

This thesis is divided into six chapters that are organised as follows. Chapter 1 starts with an introduction to the joint pricing and production planning problems, in particular multiple products over a multiperiod horizon along with its importance. Essential features and objective of this project are also discussed. Section 1.2 provides the current literature that is closely related to this project.

Chapter 2 begins by introducing the main features of the problem of time-varying pricing and production planning of multiple products over a multiperiod horizon. This problem is formulated as a nonlinear programming problem. The cases of without and with backorders are presented in Sections 2.2 and 2.3, respectively. The linear function of demand/price is discussed as a special case of the problem, which is quite common in deterministic models. An alternative formulation is to have a convex set of constraints and the problem becomes a Mixed Integer Nonlinear Programming problem over a linear set of constraints. The limitation of obtaining the optimal solution of the developed models is discussed. The Chapter concludes by describing our strategy in this thesis to overcome the above limitations to solve the models.

In Chapter 3 the aim is to tackle the main time-varying problem defined in Chapter 2 through solving an easier case when prices are constant. The Chapter begins by bringing the additional features of the problem and formulating it in a mathematical programming model. Section 3.2 presents an iterative solution strategy for the developed model, which constructs a tree searched in breadth-first manner. A detailed algorithm is presented in Section 3.3 along with its main steps flowchart. The proposed method is implemented through a small but practical size numerical example. Our numerical example shows that by utilising the proposed algorithm in this chapter, at any iteration we easily solve a small size nonlinear problem over a linear set of constraints. This brings a promising use of existing commercial packages to handle the large practical problems. In Section 3.4 we conclude by discussing about the results and achievements of the chapter.

In Chapter 4 we reformulate the problem of time-variant joint pricing and production planning in order to utilise an extension of the algorithm proposed in Chapter 3. We consider multi blocks of constant pricing problems which can lead to the time-varying pricing when each block consists of just one period. An iterative solution strategy is discussed in Section 4.2. A detailed algorithm is presented in Section 4.3 which is an extension of the algorithm proposed in Chapter 3. A practical size numerical example is used to implement the algorithm in Section 4.3. The result of the example is compared with the given formulation in Chapter 2, which shows the capability of the algorithm to obtain the same optimal solution. The significance of this algorithm is more understood for large scale problems which cannot be solved easily with the existing commercial packages.

In Chapter 5 the uncertainty of the demand/price function is incorporated into the model. This function can be chosen from a set of scenarios. The purpose of this Chapter is to develop a robust optimisation (RO) model to determine the optimal production planning and constant pricing of a manufacturing system with multiple products over a multiple period horizon to maximize the total profit. Section 5.1 briefly reviews the robust optimisation approach and its formulation in the case of a Linear Programming problem. Section 5.2 presents the deterministic model for the problem of joint pricing and production planning developed in Gilbert (2000). In this Section we propose a robust optimisation model for the joint pricing and production planning with uncertain demand/price function. Section 5.3 reviews the existing solution methods developed for nonlinear programming problems. Section 5.4 illustrates our model and its solution with two practical size examples. The importance of this Chapter is the novel use of robust optimisation in the discrete case of joint pricing and production planning.

Finally, Chapter 6 gives a summary of the earlier Chapters and conclusions obtained from the research. Also suggestions are made for future research work.

# Chapter 2

# Models

In this Chapter we consider the problem of joint pricing and production planning of multiple products over a multi-period horizon. Some important features of the problem are:

- The planning horizon consists of $T$ periods.
- The firm produces $n$ different products and the demand of each product is period-varying over the planning horizon.
- Demand of each product is deterministic and dependent on its price.
- The production capacity is limited and shared among different products.
- Products use the same amount of capacity; here each product uses one unit of capacity.
- The production set up cost is negligible.

This Chapter is organised as follows. Section 2.1 introduces the notation and terminology used in our model. In Section 2.2 the problem of joint pricing and production planning without backorders is discussed and a model is presented. A special case of the linear demand/price function is discussed. Section 2.3 incorporates backorders into the model and gives an alternative model to have a convex set of constraints which is useful for the linear case of demand/price function.

## 2.1. Notation and terminology

We make use of the following notation and terminology in our models.

**Parameters:**

- $T$ : the number of periods.
- $n$ : the number of products.
- $c_{jt}$ : the production cost of one unit of item $j$ in period $t$.
- $h_{jt}$ : the holding cost of one unit of item $j$ in inventory for one period in period $t$.
- $s_{jt}$ : the backordering cost of one unit of item $j$ for one period in period $t$.
- $K_t$ : the total amount of shared production capacity in period $t$.

**Variables:**

- $p_{jt}$ : the price of product $j$ in period $t$.
- $P$ : the $n \times T$ price matrix
- $D_{jt}$ : the demand for item $j$ in period $t$, this has been induced by the chosen price.
- $D$ : the $n \times T$ demand intensity matrix.
- $x_{jt}$ : the amount of product $j$ produced in period $t$.
- $y_{jt}$ : the amount of product $j$ held in inventory at the end of period $t$.
- $z_{jt}$ : the amount of product $j$ backordered from period $t$ to meet the demand of period $t-1$.
- $X$ : the $n \times T$ production matrix.
- $Y$ : the $n \times T$ inventory matrix.
- $Z$ : the $n \times T$ backordering matrix.

**Functions:**

- $D_{jt}(P)$: the relationship between demand and price of item $j$ in period $t$, this function has an inverse, $P_{jt}(D)$.

*$R_{jt}(D)$ :* the revenue function as $D_{jt}P_{jt}(D)$.

Note that, here the relationship between demand intensity and price is known, but both of them are decision variables of the problem.

We assume that corresponding to each demand intensity matrix, there is just one price matrix; and for each price matrix there is just one demand intensity matrix. In our work, we consider situations, in which different products are presented to different market sectors. Hence there is no interaction between the price of one product and the demand of one another product or in other words, the cross price elasticity among various products is zero. By this assumption, we have the convenience of using $D_{jt}$ for $j =1, . . .,n$ and $t=1,…,T$ as the decision variables.

In the next sections, we express the problem of jointly determining the price and production plan for two cases, first when backorders are not allowed, next with allowed backorders.

## 2.2. Joint Pricing and Production Planning for Multiple Products without Backorders

In this case, because backorders are not allowed, the price of each product in each period should be determined in a way that all induced demands be satisfied by using the shared capacity of that period and previous ones.

Gilbert (2000) formulates this problem for the case of constant pricing. He assumes that the demand for each of $n$ items is seasonal and dependent on the prices to which the firm commits for the entire planning horizon. He denotes the price for product $j$ by $p_j$, and the demand intensity for product $j$ by $D_j(p)$, where $p$ is the n-dimensional price vector. He imposes the following assumptions on these demand intensity functions:

ASSUMPTION 1. There is a one-to-one correspondence between price vectors and demand intensity vectors so that $D_j(p)$ has an inverse, which is denoted by $p_j(D)$, where $D$ is an n-dimensional demand intensity.

ASSUMPTION 2. The revenue $D_j p_j(D)$ function, denote by $R_j(D)$, for each product $j = 1, . . . , n$ is concave.

To model seasonality, he defines $D_{jt}$ to be the demand for item $j$ in period $t$, and assumes that there exist parameters $\beta_{jt}$ such that the $D_{jt} = \beta_{jt} D_j$.

In addition to determining the intensities of demand, $D_j$ for $j = 1, . . ., n$, to induce from the market, the firm must also determine a plan for satisfying the demand that is induced. He assumes that all of the items are produced on the same equipment and that there is limited capacity. Denote the amount of unused capacity in period $t$ as $x_{ot}$. The costs per unit are also constant as $c_j$ and $h_j$. The products are assumed to be indexed in decreasing order of their holding costs. He further defines $C(D)$ to be the minimum cost of satisfying the demand corresponding to the induced demand intensities $D_1, ..., D_n$. The problem of jointly determining the price and production plan is expressed as:

$$\pi = Max_{D \geq 0} \left\{ \pi(D) = \sum_{j=1}^{n} R_j \sum_{t=1}^{T} \beta_{jt} - C(D) \right\} \qquad (2.1)$$

such that

$$\sum_{j=1}^{n} \sum_{t=1}^{\tau} D_{jt} \leq \sum_{t=1}^{\tau} K_t \, , for \; \tau = 1,2, \dots , T. \qquad (2.2)$$

In (2.1),

$$= Min_{x,y} \left\{ C^P(D; x, y) = \sum_{j=1}^{n} (c_j x_{jT} + \sum_{t=1}^{T-1} (h_j y_{jt} + c_j x_{jt}) ) \right\} \qquad (2.3)$$

subject to:

$$-x_{j1} + y_{j1} = -\beta_{j1} D_j \, , for \; j = 1, \dots , n, \qquad (2.4a)$$

$$-x_{jt} - y_{jt-1} + y_{jt} = -\beta_{jt} D_j \, , for \; t = 2, \dots , T-1 \, ; \; j = 1, \dots , n, \qquad (2.4b)$$

$$-x_{jT} - y_{jT-1} = -\beta_{jT} D_j \text{ , } for \;\; j = 1, \dots, n, \qquad (2.4c)$$

$$\sum_{j=0}^{n} x_{jt} = K_t \text{ , } for \; t = 1, \dots, T, \qquad (2.5)$$

$$\sum_{t=1}^{T} -x_{0t} = \sum_{t=1}^{T} (\sum_{j=1}^{n} \beta_{jt} D_j - K_t) \text{ , } \qquad (2.6)$$

$$x_{jt} \text{ , } y_{jt} \geq 0, for \; t = 1, \dots, T \; ; \; j = 1, \dots, n. \qquad (2.7)$$

Constraint (2.2) ensures that he considers only demand intensity vectors for which there exists a feasible solution to the cost minimization sub-problem, *C(D)*. Constraint (2.4) is a set of flow balance equations that ensure that all of the induced demand is satisfied. Constraint (2.5) ensures that capacity in period *t* is sufficient to allow all of the production that is planned for all *n* items. Constraint (2.6) is a redundant constraint, but its inclusion makes the cost minimization sub-problem a network. The requirement in (2.7) that inventory be nonnegative assures that demand is satisfied in all periods *t =1, . . . ,T* with no backorders.

However, we introduce some additional and updated parameters and variables in Section 2.1 to obtain the time-variant model. The problem can be formulated as:

$$\pi = Max_{D,X,Y \geq 0} \left\{ \pi(D, X, Y) = \sum_{j=1}^{n} \sum_{t=1}^{T} (D_{jt} P_{jt}\,(D) - c_{jt} x_{jt} - h_{jt} y_{jt}\,) \right\} \qquad (2.8)$$

subject to:

$$\sum_{j=1}^{n} \sum_{t=1}^{\tau} D_{jt} \leq \sum_{t=1}^{\tau} K_t \;\; , for \; \tau = 1,2, \dots, T, \qquad (2.9)$$

$$D_{jt} = x_{jt} + y_{jt-1} - y_{jt} \text{ , } for \; t = 1, \dots, T \; ; \; j = 1, \dots, n, \qquad (2.10)$$

$$\sum_{j=1}^{n} x_{jt} \leq K_t \text{ , } for \; t = 1, \dots, T \; and \qquad (2.11)$$

$$x_{jt}, y_{jt}, D_{jt} \geq 0. \qquad (2.12)$$

Our objective function (2.8) includes the sales revenue minus production and inventory costs. Constraints (2.9) specify the available capacity over the planning horizon with no backorders allowed. Constraints (2.10) are a set of flow balance equations that ensure that all of the induced demand is satisfied.

Constraints (2.11) ensure that there is an adequate amount of capacity in period *t* to produce all *n* items based on the plan. Finally (2.12) is just the non-negativity series of constraints.

As can be seen, the problem has been formulated as a Nonlinear Programming problem over a set of linear constraints. The computational difficulty of the above problem is due to the non-linear objective function (2.8). This characteristic makes the task of developing an algorithm that guarantees a global optimal solution extremely difficult. However, for the case of linear demand/price function we can utilise the existing developed algorithms that yield the global optimum.

### 2.2.1. Case of linear Demand/Price function

In this section the function of the price in terms of demand, $P_{jt}(D)$, is expressed as:

$$P_{jt}(D) = a_{jt} - b_{jt}D_{jt}$$

As a result the objective function (2.8) becomes a quadratic expression of $D_{jt}$ s:

$$\pi = Max_{D,X,Y\geq 0}\left\{\sum_{j=1}^{n}\sum_{t=1}^{T}(D_{jt}\left(a_{jt} - b_{jt}D_{jt}\right) - c_{jt}x_{jt} - h_{jt}y_{jt}\ )\right\}. \tag{2.13}$$

So, the problem becomes a Quadratic Programming problem which has only linear constraints. Since a Quadratic Programming problem is a special case of a smooth nonlinear problem, it can be solved by a smooth nonlinear optimization method. However, a faster and more reliable way to solve a QP problem is to use an extension of the Simplex method or an extension of the Interior Point or Barrier method in developed commercial packages (for example CPLEX).

## 2.3. Joint Pricing and Production Planning for Multiple Products with Backorders

In the optimal solution of this problem, we impose that if in any period, there is a shortage of capacity to produce a specific type of item, there cannot be an excess capacity to produce any of all type of items.

In this case, the problem can be formulated as:

$$\pi = Max_{D,X,Y,Z\geq 0}\{\pi(D,X,Y,Z) = \sum_{j=1}^{n}\sum_{t=1}^{T}(D_{jt}P_{jt}(D) - c_{jt}x_{jt} - h_{jt}y_{jt} - s_{jt}z_{jt})\} \quad (2.14)$$

subject to:

$$\sum_{j=1}^{n}\sum_{t=1}^{T} D_{jt} \leq \sum_{t=1}^{T} K_t, \quad (2.15)$$

$$D_{jt} = x_{jt} + y_{jt-1} + z_{jt+1} - y_{jt} - z_{jt}, for\ t = 1, \dots, T\ and\ j = 1, \dots, n \quad (2.16)$$

$$\sum_{j=1}^{n} x_{jt} \leq K_t, for\ t = 1, \dots, T \quad (2.17)$$

$$y_{it}z_{jt+1} = 0, for\ t = 1, \dots, T\ and\ i, j = 1, \dots, n, \quad and \quad (2.18)$$

$$x_{jt}, y_{jt}, z_{jt}, D_{jt} \geq 0. \quad (2.19)$$

Our objective function (2.14) includes the sales revenue minus production, inventory and backordering costs. Constraint (2.15) specifies the available capacity over the planning horizon. Constraints (2.16) are a set of flow balance equations that ensure that all of the induced demand is satisfied. Constraints (2.17) ensure that there is an adequate amount of capacity in period *t* to produce all *n* items based on the plan. The requirement in (2.18) that inventory and shortage as a cross product should be zero ensures that when there is an insufficient amount of capacity in one period from *t =1, . . . ,T* the priority is to meet the demand of the same period instead of the other periods. Finally (2.19) is just the non-negativity series of constraints.

This problem is computationally difficult, because of the non-linear objective function (2.14) as well as the constraint set (2.18). Again, these characteristics make the task of developing an algorithm that guarantees the global optimal solution extremely difficult. However, there are some commercial packages available for solving Nonlinear Programming problems to local optima. For very large applications, the developed tools may even fail to converge to a feasible solution. Here by introducing some binary variables, we reformulate the problem as a Mixed Integer Nonlinear Programming problem over a linear set of constraints. By this reformulation, we have the option of solving a series of simpler sub-problems in large-scale applications. Although, for a small to medium size case, we still have the ease of using developed packages (e.g. MAPLE) to obtain the local optima.

### 2.3.1. Mixed Integer Nonlinear Programming over a Convex Set of Constraints

In order to tackle the complexity of the problem, we reformulate the non-linear constraints (2.18) as linear constraints. Corresponding to each $y_{it}z_{jt+1} = 0$ constraint, we introduce two binary variables as:

$$u_{ijt} = \begin{cases} 1, if\ y_{it} > 0 \\ 0, if\ y_{it} = 0 \end{cases} \quad and\ v_{ijt} = \begin{cases} 1, if\ z_{jt+1} > 0 \\ 0, if\ z_{jt+1} = 0 \end{cases}$$

Then the non-linear constraint can be written as:

$$u_{ijt} + v_{ijt} \leq 1$$

$$y_{it} - \varepsilon u_{ijt} \geq 0$$

$$y_{it} - M u_{ijt} \leq 0$$

$$z_{jt+1} - \varepsilon v_{ijt} \geq 0$$

$$z_{jt+1} - M v_{ijt} \leq 0$$

where, $\varepsilon$ and $M$ are very small and very big values respectively.

As a result the problem can be formulated as:

$$\pi = Max_{D,X,Y,Z\geq 0}\left\{\sum_{j=1}^{n}\sum_{t=1}^{T}(D_{jt}P_{jt}\,(D) - c_{jt}x_{jt} - h_{jt}y_{jt} - s_{jt}z_{jt}\,)\right\} \tag{2.20}$$

subject to:

$$\sum_{j=1}^{n}\sum_{t=1}^{T} D_{jt} \leq \sum_{t=1}^{T} K_t, \tag{2.21}$$

$$D_{jt} = x_{jt} + y_{jt-1} + z_{jt+1} - y_{jt} - z_{jt}\,, for\ t = 1,\dots,T\ and\ j = 1,\dots,n \tag{2.22}$$

$$\sum_{j=1}^{n} x_{jt} \leq K_t\,, for\ t = 1,\dots,T \tag{2.23}$$

$$\begin{cases} u_{ijt} + v_{ijt} \leq 1 \\ y_{it} - \varepsilon u_{ijt} \geq 0 \\ y_{it} - M u_{ijt} \leq 0 \\ z_{jt+1} - \varepsilon v_{ijt} \geq 0 \\ z_{jt+1} - M v_{ijt} \leq 0 \end{cases}, for\ t = 1,\dots,T\ and\ i,j = 1,\dots,n \tag{2.24}$$

$$u_{ijt}\ and\ v_{ijt}\ are\ binary\ variables, for\ t = 1,\dots,T\ and\ i,j = 1,\dots,n. \tag{2.25}$$

Now this problem is a Mixed Integer Nonlinear Programming problem over a convex set of constraints. If the demand/price function is linear, this model can be utilised to formulate the problem as a Mixed Integer Quadratic Programming problem with all linear constraints and some binary variables.

As we discussed earlier in this Chapter, due to the nonlinearity both cases of with and without backorders lead to computational difficulties in finding a solution. As a result, in the following Chapters we look for another solution strategy to tackle the problem in an easier way by eliminating the nonlinear constraints while keeping the track of feasibility. As the case of without backorders has already been addressed by Gilbert (2000) for constant pricing, we only consider the more complex problem with allowed backorders in Chapters 3 and 4. In Chapter 3 we formulate the constant pricing problem with backorders and present a detailed algorithm to solve the model. In Chapter 4, by extending the algorithm, the case of time-variant pricing is considered. A practical size numerical example is given and the solution of the developed model is compared to the formulation presented in Chapter 2. The significance of the algorithms in Chapters 3 and 4 is more understood for large scale problems which cannot be solved easily with the existing commercial packages.

# Chapter 3

# Joint Pricing and Production Planning for Fixed Priced Multiple Products with Backorders

In this Chapter, we investigate the case where backorders are allowed and the excess demand of products in some periods can be backlogged if it is more profitable. The application that motivated this research is manufacturing pricing, where the products are non-perishable assets and can be stored to fulfil the future demands. We assume that the firm is not flexible to change the price list frequently and usually has long-term contracts with Original Equipment Manufacturers (OEMs). Additionally, in some companies, the price announcement to market is done by publishing the price lists which cannot be adjusted easily. Hence the price change will bring a considerable cost to them. In general, choosing a constant price over a finite horizon facilitates the maintenance of a stable set of loyal customers.

Our problem is computationally difficult, because it involves nonlinear objective function and some nonlinear constraints. Our strategy to reduce the level of difficulty is to utilise a method that solves the relaxed problem which considers only linear constraints. However, our method keeps track of the feasibility with respect to the nonlinear constraints in the original problem. The developed model which is a combination of Linear Programming (LP) and Nonlinear Programming (NLP) is solved iteratively.

Iterations consist of two main stages. The first stage starts with a known basis for the LP, solves the linear equations corresponding to the chosen basis and finds the value of the LP's objective function in terms of the main problems' decision variables (which is the demand intensity of each product induced by the pricing policy). The second stage receives the output of the first stage and based on that, finds the structure of the NLP. Next, the corresponding linear constraint set to the chosen basis in stage 1 is defined and the NLP is solved subject to the determined linear constraint set. Depending on the result of the NLP solution, some candidate bases will be revealed to restart iteration and repeat stages 1 and 2. To achieve the final optimal solution, a branching type procedure is utilised which will stop given that all next level branches have been visited in earlier iterations. Bearing in mind the fact that the backorder case makes the problem computationally difficult, our proposed strategy is practical to implement, as we demonstrate through a numerical example.

This Chapter is organised as follows. In Section 3.1 we present the problem features and formulate it in a mathematical programming model. Section 3.2 explains a solution strategy for the developed model and detailed algorithms along with implementation are presented in Section 3.3 through a numerical example.

## 3.1. Model

This Section formulates the problem of joint fixed pricing and production planning of multiple products with allowable inventory carrying and backorders. We first recall the following features of the problem, which listed in Chapter 2.

- The planning horizon consists of *T* periods.
- The firm produces *n* different products and the demand of each product is period-varying and seasonal over the planning horizon.
- Demand of each product is deterministic and dependent on its price.

- The production capacity is limited and shared among different products.
- Products use the same amount of capacity; here each product uses one unit of capacity.
- The production set up cost is negligible.

In this chapter we consider the following additional features:

- The price of each product is constant over the total planning horizon.
- All related costs including production, holding and shortage costs are constant over the planning horizon for each product.

We make use of the following notation and terminology in the description of our model.

**Parameters:**

$n$ : the number of products

$T$ : the number of periods

$c_j$ : the production cost of one unit of item $j$; $j=1,2,\dots,n$

$h_j$ : the holding cost of one unit of item $j$ in inventory for one period

$s_j$ : the backordering cost of one unit of item $j$ for one period

$K_t$ : the total amount of available capacity in period $t$

$\beta_{jt}$ : the seasonality parameter of item $j$ in period $t$

**Variables:**

$p_j$ : the price of product $j$; $j=1,2,\dots,n$

$p$ : the n-dimensional price vector

$D$ : the n-dimensional demand intensity vector

$D_{jt}$ : the demand for item $j$ in period $t$; $j=1,2,\dots,n$ and $t=1,2,\dots,T$

$x_{jt}$ : the amount of product $j$ produced in period $t$

$y_{jt}$ : the amount of product $j$ held in inventory at the end of period $t$

$z_{jt}$ : the amount of product $j$ backordered from period $t$ to meet the demand of period $t-1$

$x_{0t}$ : the amount of unused capacity in period $t$

$X$ : the n×T production matrix

$Y$ : the n×T inventory matrix

$Z$ : the n×T backordering matrix

**Functions:**

$D_j(p)$ : the demand intensity for product $j$, which is a function of price vector

Note that the relationship between demand intensity and price is known, but both of them are decision variables of the problem.

$R_j(D)$ : the revenue function as $D_j(p).\ p_j$

$C(D)$ : the minimum cost of satisfying the demand corresponding to the induced demand intensities $D_1, \ldots, D_n$

We assume that corresponding to each demand intensity vector, there is just one price vector; and for each price vector there is just one demand intensity vector. In this study, we look at situations, in which different products are presented to different market sectors. Hence there is no interaction between the price of one product and the demand of other products or in other words, the cross price elasticity among various products is zero. By this assumption, we have the ease of using $D_j$ for $j = 1, \ldots, n$ as the decision variables. The other assumption relies on the concavity of the revenue function, $R_j(D)$, for each product $j = 1, \ldots, n$. The seasonality model is assumed to be a purely multiplicative and so $D_{jt} = \beta_{jt} \cdot D_j(p)$. We can explain this assumption by considering that the distribution of price sensitivity among the participants in the market doesn't change although the size of the market may differ in different periods. This justification is an interpretation of the model used for a single product in Gilbert(1999). On the contrary, Kunreuther and Schrage (1973) for the single product model assume a price-insensitive additive seasonality term with the intention that demand in period $t$ is expressed as $d_t(p) = \alpha_t + \beta_t D(p)$. Although this is more general than the purely multiplicative model, we note that in the application of their model, Kunreuther and Schrage (1973) assume that seasonality is purely additive, i.e., $\beta_1=\beta_2=\ldots\beta_T$. As well, we assumed that the products are indexed in decreasing order of their holding and shortage costs, i.e., $h_i \geq h_j$ and $s_i \geq s_j$ for $i<j$. Like any other inventory system, the shortage cost is always more than the holding cost.

The problem of jointly determining the price and production plan can be formally expressed as follows:

$$\pi = Max_{D\geq 0} \left\{\pi(D) = \sum_{j=1}^{n} R_j(D) \sum_{t=1}^{T} \beta_{jt} - C(D)\right\} \quad (3.1)$$

such that

$$\sum_{j=1}^{n} \sum_{t=1}^{T} D_j \beta_{jt} \leq \sum_{t=1}^{T} K_t. \quad (3.2)$$

In (3.1),

$$C(D) = Min_{X,Y,Z\geq 0} \left\{\sum_{j=1}^{n} \sum_{t=1}^{T} (c_j x_{jt} + h_j y_{jt} + s_j z_{jt})\right\} \quad (3.3)$$

subject to:

$$D_{jt} = x_{jt} + y_{jt-1} + z_{jt+1} - y_{jt} - z_{jt}\ , for\ t = 1, \ldots, T\ and\ j = 1, \ldots, n, \quad (3.4)$$

$$\sum_{j=0}^{n} x_{jt} = K_t\ , for\ t = 1, \ldots, T, \quad (3.5)$$

$$y_{it} z_{jt+1} = 0\ , for\ t = 1, \ldots, T\ and\ i, j = 1, \ldots, n, \quad (3.6)$$

$$x_{jt}, y_{jt}, z_{jt}, D_{jt} \geq 0, for\ t = 1, \ldots, T\ and\ j = 0, \ldots, n. \quad (3.7)$$

We refer to the problem (3.3)-(3.7) as the Cost Minimization Sub Problem, "CMSP".

The objective function (3.1) consists of the sales revenue minus the total cost associated with the chosen demand intensity. Constraint (3.2) ensures that only demand intensity vectors which result in a feasible solution to the CMSP have been considered. Constraints (3.4) are a set of flow balance equations that ensure that all of the induced demand is satisfied. Constraints (3.5) ensure that there is an adequate amount of capacity in period $t$ to produce all $n$ items based on the plan. The requirement in (3.6) that inventory and shortage as a cross product should be zero ensures that when there is an insufficient amount of capacity in one period from $t = 1, \ldots, T$ the priority is to meet the demand of the same period instead of the others' periods. Finally (3.7) is just the non-negativity series of constraints.

For the case when backorders are not allowed, Gilbert (2000) utilised the fact that the objective function to be maximized in the model was concave in the demand vector $D = [D_1, \ldots, D_n]$. This property holds also for the backorder case in our model. Using an argument similar to Gilbert we established that:

THEOREM 1. The profit function, $\pi(D)$ that is to be maximized in (3.1) is concave in the demand vector.

PROOF. Take two demand vectors $D^1$ and $D^2$, and let $D^3 = \alpha D^1 + (1-\alpha) D^2$ where $\alpha \in (0, 1)$. Let $(x^1, y^1, z^1)$ and $(x^2, y^2, z^2)$ be the optimal solution variables associated with $C(D^1)$ and $C(D^2)$ respectively. Thus, $(x^1, y^1, z^1)$ and $(x^2, y^2, z^2)$ are feasible with respect to Constraints (3.4-3.7) when $D^1$ and $D^2$ are in the left-hand side of (3.4) respectively. Let $\tilde{x} = \alpha x^1 + (1-\alpha)x^2$, $\tilde{y} = \alpha y^1 + (1-\alpha)y^2$ and $\tilde{z} = \alpha z^1 + (1-\alpha) z^2$. Clearly, $(\tilde{x}, \tilde{y}, \tilde{z})$ satisfies all of the constraints in (3.4-3.7). Therefore:

$$
\begin{aligned}
&\alpha\pi(D^1) + (1-\alpha)\pi(D^2) \\
&= \sum_{j=1}^{n}(\alpha R_j(D^1) + (1-\alpha)R_j(D^2)) \sum_{t=1}^{T} \beta_{jt} - \alpha C(D^1; x^1, y^1, z^1) \\
&\quad + (1-\alpha)C(D^2; x^2, y^2, z^2) \\
&= \sum_{j=1}^{n}(\alpha R_j(D^1) + (1-\alpha)R_j(D^2)) \sum_{t=1}^{T} \beta_{jt} - C(D^3; \tilde{x}, \tilde{y}, \tilde{z}), \qquad (3.8)
\end{aligned}
$$

where the latter equality follows from the linearity of the objective function in (3.3) and the fact that $(\tilde{x}, \tilde{y}, \tilde{z})$ is a feasible solution to (3.4-3.7) when $D^3$ is in the left-hand side of (3.4). By assumption, $R_j(D)$ is concave. Therefore, α $R_j(D^1)$+ (1-α) $R_j(D^2) \leq R_j(D^3)$ for each $j = 1, \ldots, n$. By substituting for each term in the summation of the right-hand side of (3.8) , we have:

$$\alpha\pi(D^1) + (1-\alpha)\pi(D^2) \leq \sum_{j=1}^{n} R_j(D^3) \sum_{t=1}^{T} \beta_{jt} - C(D^3; \tilde{x}, \tilde{y}, \tilde{z}) = \pi(D^3) \quad (3.9)$$

where the latter inequality results from the principle of optimality and the definition of $C(D^3)$ as the optimal (minimum) cost of satisfying demand $D^3$. □

## 3.2. Solution Strategy

Given the capacity limitations and other parameters, the firm must decide upon production quantities, inventory and backorder levels for each item as well as a constant price at which it commits to sell the products over the total planning horizon.

Note that for each *D* vector, as the decision variable of the model, there is an optimal solution to the CMSP. In other words, when the *D* vector is changed the coefficients of the profit function, *π(D),* will also change. Consequently, the problem can be solved iteratively. Each iteration starts with a known basis for the CMSP and involves two stages:

**Stage 1: Solve the cost minimization sub problem, CMSP**

1) Consider a known basis, *B(D)*, for the CMSP.
   Each basis consists of some of the *x,y* and *z* variables for $j=0,1,\ldots,n$ and $t=1,2,\ldots,T$.
2) Find the values of the basic variables for CMSP in terms of *Dj's*.
   This step can be done by using equations (3.4) and (3.5).
3) Find the value of the objective function of CMSP, *C(D)*, in terms of *Dj's* by using (3.3).

**Stage 2: Solve the main Non-Linear problem and update the basis**

1) Restructure the profit function, $\pi(D)$, subject to above defined $C(D)$ function and using (3.1).

2) Determine a linear constraint set $\Omega(B(D))$ such that $B(D)$ is an optimal basis for any $D' \in \Omega(B(D))$.

Concerning the Theorem 1, we would guarantee to obtain an optimal solution for the concave objective function, $\pi(D)$, over a linear set of constraints, $\Omega(B(D))$.

Note that once $B(D)$ is determined, $\Omega(B(D))$ can be fully specified by using constraint set generation procedure presented in Section 3.2.1.

3) Solve the *NLP,* that maximizes $\pi(D)$ subject to $D \in \Omega(B(D))$.

4) If any of the constraints are binding in the optimal solution to this *NLP*, determine how the basis for the CMSP should change. Find the candidate new basis.

The procedure to do step 4 is mentioned in Section 3.2.2 by details.

### 3.2.1. Constraint set, $\Omega(B(D))$, generation procedure.

**Input:**

The *x,y* and *z* basic variables.

**Output:**

A set of linear constraints, $\Omega(B(D))$ , $\Omega(B(D)) = A_1 U\, A_2 U\, A_3$, in terms of $D_j$'s.

**Steps:**

1) Define $i(t)$ as the number of $x_{jt}$ variables in the basis for $j=0,\ldots,n$.

2) For each period $t$, if $x_{0t} \in B(D)$, calculate the following constraint in terms of $D_j$'s.

$$\sum_{j=1}^{n} x_{jt} \leq K_t. \qquad (3.10)$$

3) Add (3.10) constraint in a set named $A_1$.

4) If $x_{0t} \notin B(D)$ and $y_{i(t)t-1} \in B(D)$ , calculate the following constraints in terms of $D_j$'s.

$y_{i(t)t-1} > 0$ (3.11a)

$x_{i(t)t} \geq 0.$ (3.11b)

5) If $x_{0t} \notin B(D)$ and $y_{i(t)t-1} \notin B(D)$ , calculate the following constraints in terms of $D_j$'s.

$z_{i(t)t+1} > 0$ (3.11a)

$x_{i(t)t} \geq 0.$ (3.11b)

6) Add (3.11a) constraint in a set named $A_2$ and (3.11b) in a set named $A_3$.

7) Define $\Omega(B(D)) = A_1 U A_2 U A_3$.

## 3.2.2. Finding new candidate basis procedure.

**Input:**

Solution of *NLP* and list of binding constraint in $\Omega(B(D)) = A_1 U A_2 U A_3$.

**Output:**

Candidates for a change of the current basis, *B(D)*.

Before going to the main steps of the procedure we need to define some values as follow:

$$\theta(t,j): Min\left\{t' \geq t;\ y_{jt'} \notin B(D)\right\}$$

$$\alpha(t,j): Max\left\{t' \leq t;\ z_{jt'} \notin B(D)\right\}$$

$$\tau(t): Min\left\{\alpha(t,j);\ j = 1, \ldots, n\right\}$$

$$a(t): \left\{j: \alpha(t,j) = \tau(t)\ and\ s_j - h_j \geq s_i - h_i\ for\ i \neq j\right\}$$

$$\tau'(t): Max\left\{\theta(t,j);\ j = 1, \ldots, n\right\}$$

$$b(t): \left\{j: \theta(t,j) = \tau'(t)\ and\ s_j - h_j \leq s_i - h_i\ for\ i \neq j\right\}.$$

**Steps:**

1) For any binding constraint which belongs to set $A_2$, the new candidates can be determined as follows:

- $If\ \left(z_{i(t)t+1} \in B(D), for\ j = 1, \dots, n\right) \left\{first\ candidate\ basis = B(D) - \{z_{i(t)t+1}\} + \{x_{0t}\ or\ x_{i(t)+1\ t}\}\right\}$
- $Else\ \left\{first\ candidate\ basis = B(D) - \{y_{i(t)t-1}\} + \{x_{0t}\ or\ x_{i(t)+1\ t}\}\right\}$.

2) For any binding constraint which belongs to set $A_3$, the new candidates can be determined as follows:

**Case 1:** ***t=T***

$\left\{first\ candidate\ basis = B(D) - \{x_{i(t)t}\} + \{y_{i(t)-1\ t-1}\}\right\}$.

**Case 2:** **1<*t*<*T***

$\left\{first\ candidate\ basis = B(D) - \{x_{i(t)t}\} + \{y_{i(t)-1\ t-1}\}\right\}$;

- $If\ \left(y_{jt} \notin B(D), for\ j = 1, \dots, n\right) \left\{second\ candidate\ basis = B(D) - \{x_{i(t)t}\} + \{z_{i(t)-1\ t+1}\}\right\}$.

**Case 3:** ***t=1***

$\left\{first\ candidate\ basis = B(D) - \{x_{i(t)t}\} + \{z_{i(t)-1\ t+1}\}\right\}$.

3) For any binding constraint which belongs to set $A_1$, the new candidates can be determined as follows:

**Case 1:** ***t=T***

$$If\ \left(z_{jt} \notin B(D), for\ j = 1, \dots, n\right)\{first\ candidate\ basis = B(D) - \{x_{0T}\} + \{y_{nT-1}\}\}$$

$$Else\ \left\{first\ candidate\ basis = B(D) - \{z_{a(t)\tau(t)+1}\} + \{y_{a(t)\tau(t)-1}\}\right\}.$$

**Case 2:** ***t =T-1***

$If\ (z_{jt} \notin B(D), for\ j = 1, \dots, n)\{first\ candidate\ basis$
$= B(D) - \{x_{0t}\} + \{y_{nt-1}\};$

- $If\ (y_{jt} \notin B(D), for\ j = 1, \dots, n)\{second\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{n\,t+1}\}\}\}$

$Else\ \{$

- $If\ (y_{jt} \notin B(D), for\ j = 1, \dots, n)\{first\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{n\,t+1}\};$
    - $If\ (\tau(t) > 1)\{second\ candidate\ basis = B(D) - \{z_{a(t)\tau(t)+1}\} + \{y_{a(t)\tau(t)-1}\}\}\}$
- $Else\ \{first\ candidate\ basis = B(D) - \{z_{a(t)\tau(t)+1}\} +\}\}.$

**Case 3:** ***2<t<T-1***

$If\ (z_{jt} \notin B(D), for\ j = 1, \dots, n)\{first\ candidate\ basis$
$= B(D) - \{x_{0t}\} + \{y_{nt-1}\};$

- $If\ (y_{jt} \notin B(D), for\ j = 1, \dots, n)\{second\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{n\,t+1}\}\}\}$
- $Else\ \{$
    - $If\ (\tau'(t) < T)\{second\ candidate\ basis = B(D) - \{y_{b(t)\tau'(t)-1}\} + \{z_{b(t)\tau'(t)+1}\}\}\}$

$Else\ \{$

- $If\ (y_{jt} \notin B(D), for\ j = 1, \dots, n)\{first\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{n\,t+1}\};$

- $If\ (\tau(t) > 1)\left\{second\ candidate\ basis = B(D) - \left\{z_{a(t)\tau(t)+1}\right\} + \left\{y_{a(t)\tau(t)-1}\right\}\right\}$

- $Else\ \{$
  - $If\ (\tau(t) > 1)\{first\ candidate\ basis = B(D) - \left\{z_{a(t)\tau(t)+1}\right\} + \left\{y_{a(t)\tau(t)-1}\right\};$
    - $If\ (\tau'(t) < T)\left\{second\ candidate\ basis = B(D) - \left\{y_{b(t)\tau'(t)-1}\right\} + \left\{z_{b(t)\tau'(t)+1}\right\}\right\}\right\}$
  - $If\ (\tau(t) = 1)\left\{first\ candidate\ basis = B(D) - \left\{y_{b(t)\tau'(t)-1}\right\} + \left\{z_{b(t)\tau'(t)+1}\right\}\right\}\right\}.$

**Case 4:** ***t=2***

$If\ \left(z_{jt} \notin B(D), for\ j = 1, \dots, n\right)\{first\ candidate\ basis$
$$= B(D) - \{x_{0t}\} + \{y_{nt-1}\};$$

- $If\ \left(y_{jt} \notin B(D), for\ j = 1, \dots, n\right)\left\{second\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{n\ t+1}\}\right\}\right\}$
- $Else\ \{$
  - $If\ (\tau'(t) < T)\left\{second\ candidate\ basis = B(D) - \left\{y_{b(t)\tau'(t)-1}\right\} + \left\{z_{b(t)\tau'(t)+1}\right\}\right\}\right\}$

$Else\ \{$

- $If\ \left(y_{jt} \notin B(D), for\ j = 1, \dots, n\right)\left\{first\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{n\ t+1}\}\right\}\right\}$
- $Else\ \left\{first\ candidate\ basis = B(D) - \left\{y_{b(t)\tau'(t)-1}\right\} + \left\{z_{b(t)\tau'(t)+1}\right\}\right\}\right\}.$

**Case 5:** ***t=1***

$$If\ \left(y_{jt} \notin B(D), for\ j = 1, \dots, n\right)\{first\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{nt+1}\}\}$$

$$Else\ \left\{first\ candidate\ basis = B(D) - \left\{y_{b(t)\tau'(t)-1}\right\} + \left\{z_{b(t)\tau'(t)+1}\right\}\right\}.$$

## 3.3. Implementation

In this Section, two algorithms are presented to implement the solution strategy: Semi-Random and Greedy. Because of the similarities between two algorithms, just the Semi-Random is discussed in detail.

### 3.3.1. Semi-Random algorithm "SRA"

In this algorithm after the initialization step, which starts with a specific basis and proposes some other candidates to change the specific basis, the suggested candidates will be visited level by level. In other words, in each level, all the proposed candidates will be checked completely before going to the next level. In fact, we explore the search tree in a breadth-first search (BFS) manner.

In order to bring the algorithm's steps by detail, we need to define some more parameters and variables as well as previously defined:

**Parameters**

*$P_j(D_j)$ :* the price function of product $j$ , which is inverse of demand intensity function. For instance: $P_j\left(D_j\right) = a_j - b_j D_j$.

**Variables**

*B(D) :* set of $x_{jt}$, $y_{jt}$ and $z_{jt}$ variables chosen as basic variables, "basis".

*B'(D) :* as an indicator of a basis that is being currently tested.

*Ω(B(D)) :* set of linear constraints induced by *B(D)* in terms of *$D_j$s.* This is defined by the procedure detailed in Section 3.2.1.

*C(D) :* cost minimization objective function, which is formulated as:

$$\sum_{j=1}^{n}\sum_{t=1}^{T}(c_j x_{jt} + h_j y_{jt} + s_j z_{jt}).$$

*π(D)* : non-linear problem's objective function, which is structured as:

$$\pi(D) = \sum_{j=1}^{n} P_j(D_j) D_j \sum_{t=1}^{T} \beta_{jt} - C(D).$$

*LCS :* set of linear constraints as follows:

$$\begin{cases} D_{jt} = x_{jt} + y_{jt-1} + z_{jt+1} - y_{jt} - z_{jt}, for\ t = 1, \dots, T\ and\ j = 1, \dots, n, \\ \sum_{j=0}^{n} x_{jt} = K_t, for\ t = 1, \dots, T. \end{cases}$$

*V :* set of visited basis

Note that every basis in this set has a *π(D)* value , which can be obtained by solving the *NLP* subject to that basis.

*$U_i$ :* set of non-visited basis in the $i^{th}$ iteration.

*POS :* matrix of potential optimal solutions *($D^*$)* along with the corresponding basis and objective function value, *π(D)*.

$$POS = \begin{pmatrix} D^* & B(D) & \pi(D) \\ \vdots & \vdots & \vdots \\ D_i^* & B_i(D) & \pi_i(D) \end{pmatrix}$$

**Main body**

Initialize by *i=0 and B(D) = B'(D) = V = $U_i$ = POS =* $\phi$

1. *B'(D)={$x_{jt}$; j=0,…,n and t=1,…,T}= $B_0$(D).*

   If a variable $\notin$ *B'(D)*, it means that it has a zero value.

2. Calculate $x_{jt}$,$y_{jt}$ and $z_{jt}$ values in terms of *$D_j$s* by using the *LCS*. Calculate the *C(D)* value for the above found $x_{jt}$, $y_{jt}$ and $z_{jt}$ in terms of *$D_j$s* by using its formulation. Define the *π(D)* in terms of *$D_j$s* by using its structure definition.

3. Apply the procedure defined in Section 3.2.1 to create the set of constraints, *Ω'(B(D))* induced by B'(D).
4. Solve the nonlinear problem of:

    Maximise *π(D),* structured in step 2,

    subject to:

    *Ω'(B(D))* defined in step 3 and

    $\sum_{j=1}^{n}\sum_{t=1}^{T} D_j\beta_{jt} \leq \sum_{t=1}^{T} K_t.$

    This step is in fact optimizing a nonlinear concave objective function over a linear constraint set which guarantees achieving an optimal solution.
5. Add *B'(D)* and it's corresponding *π(D)* to *V.* i.e.*, V=V+B'(D).*
6. Check whether or not there is any binding constraint in *Ω'(B(D))* subject to *D**. If NO, *D** is the optimal solution of the problem. Go to step 13. If YES, put *i=i+1* and go to step 7.
7. Apply the procedure defined in Section 3.2.2 to find the candidate basis for each binding constraint and label them as $B_1(D)$, $B_2(D)$, $B_3(D)$, …

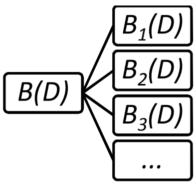



*Figure 3.1. Labelling the found candidate bases*

Note that the parent of all candidates found is the basis which was recently being tested, *B'(D)*.

8. Add all candidates found in step 7 to $U_i$.

$$U_i = \{B_1(D), B_2(D), B_3(D), \dots\}$$

9. Choose one of the basis in set $U_i$ at random; suppose that it is $B_c(D)$. For the chosen basis, do the following steps:

    9.1. $B'(D)=B_c(D)$

    9.2. Repeat steps 2 and 3 for this new basis, *B'(D)*.

9.3. If the created *Ω'(B(D))* has feasible area, repeat step *4* for this new basis and go to step 9.4. Otherwise

*9.3.1.* Add parent of *B'(D)* and it's corresponding *D** and *π(D)* to *POS.*

*9.3.2.* Add *B'(D)* to *V,* give a zero value to the basis in set *V,* delete *B'(D)* from $U_i$ , *i=i+1* and go to step 10 .

9.4. Add *B'(D)* and it's corresponding *π(D)* to *V,* delete *B'(D)* from $U_i$.

9.5. Check whether or not there is any binding constraint in *Ω'(B(D))* subject to *D**. If NO, Add *B'(D)* and it's corresponding *D** and *π(D)* to *POS. i=i+1*, go to step 10. If YES, *i=i+1.*

9.6. Apply the procedure defined in Section 3.2 to find the candidate basis for each binding constraint and label them as $B_{c1}(D)$, $B_{c2}(D)$, $B_{c3}(D)$, …

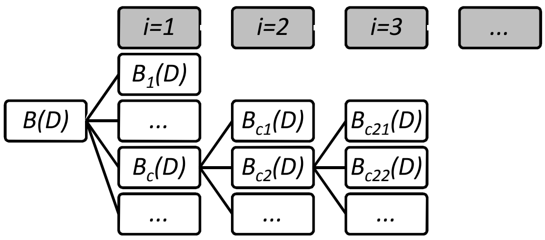



*Figure 3.2. Labelling the found candidate bases for each binding constraint*

Note that the number of index in *B(D)* is equal to *i*. For example in the first iteration we have $B_1(D)$, $B_2(D)$, $B_3(D)$, … , in the second iteration we have $B_{11}(D)$, $B_{12}(D)$, $B_{13}(D)$, … and in the fifth iteration we have $B_{12312}(D)$, $B_{21342}(D)$, $B_{42315}(D)$, …

9.7. Add those candidates found in step 9.6 which are not equal to none of the elements of set *V* and $U_1,U_2,\ldots,U_i$ to $U_i$ . Update *POS* by those candidates found in step 9.6 which are already in set *V or* $U_1,U_2,\ldots,U_i$ to $U_i$.

10. *i=i-1* and restart from step *9*; continue till set $U_i$ comes to $\phi$ .

11. *i=i+1* , check if $U_i$ exists and is not $\phi$ go to step 9, if it doesn't exist or is $\phi$ go to 12.

12. Find the biggest value of *π(D)* in *POS*, and show the corresponding $D^*$ and basis for that.

13. Calculate the optimal price for each product by using the price function, $P_j(D_j)$. Show the values of x, y, z variables corresponding to the optimal basis.

    The following flowchart summarises the main steps of the algorithm.

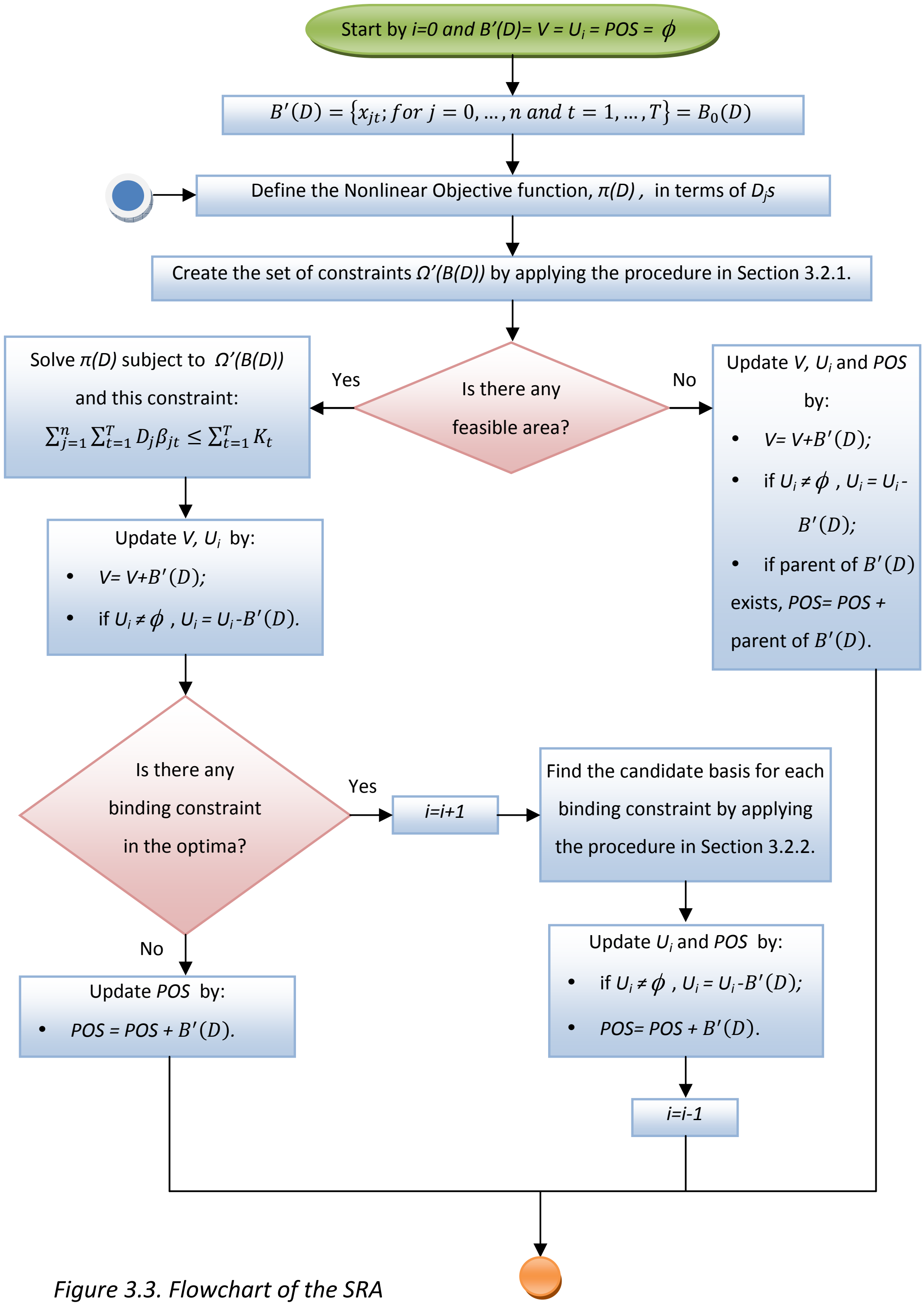



*Figure 3.3. Flowchart of the SRA*

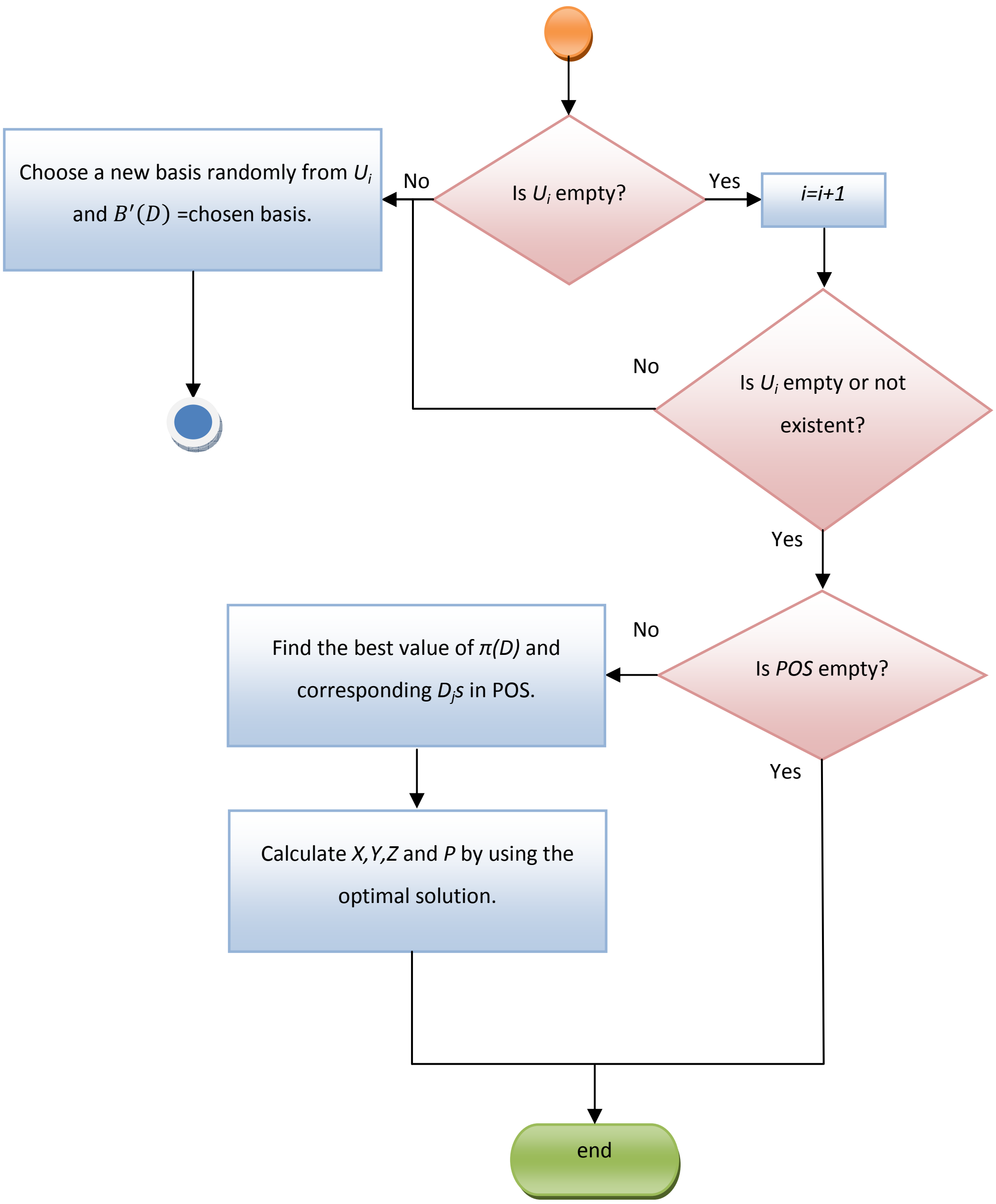



*Figure 3.3b. Flowchart of the SRA (continue)*

## 3.3.2. Greedy algorithm "GA"

In this algorithm, the only important dissimilarity with the Semi-Random algorithm is that we don't need to define set $U_i$ in each level as the set of non-visited basis. Regardless of the level of the visited basis, each basis which has a greater value for the objective function will be chosen earlier to find its next level.

### 3.3.3. Numerical Example

In order to display our algorithm more clearly, consider a case with *n=2* products and *T =6* periods. The parameters for the two products are as shown in Table 3.1.

*Table 3.1. Parameters of the example with n=2 products and T =6 periods*

| Product | $P_j(D_j)$ | $h_j$ | $s_j$ | $c_j$ | $\beta_{j1}$ | $\beta_{j2}$ | $\beta_{j3}$ | $\beta_{j4}$ | $\beta_{j5}$ | $\beta_{j6}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| **j=1** | $30-0.2\,D_1$ | 6 | 8 | 0 | 0.6 | 0.5 | 0.2 | 3 | 1.5 | 0.2 |
| **j=2** | $30-0.2\,D_2$ | 2.5 | 4 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

Note that the relationship between price and demand intensity for both products is identical, and that their cross price elasticity is zero. For ease of manual computation, we have specified the unit costs to be zero; however nonzero production cost can be covered by this algorithm. Finally, we assume a fixed production capacity, $K_t$=I40, for all periods in the planning horizon.

In order to solve this example we choose the Semi-Random algorithm and follow it step by step:

Initialize by *i=0 and B(D) = B'(D) = V = U*$_i$ *= POS =* $\phi$ .

**1.** $B(D)=\{x_{01},x_{02},x_{03},x_{04},x_{05},x_{06},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26}\}$ and

*B'(D)=B(D).*

$x_{11}=0.6D_1, x_{12}=0.5D_1, x_{13}=0.2D_1, x_{14}=3D_1, x_{15}=1.5D_1, x_{16}=0.2D_1, x_{21}=D_2, x_{22}=D_2, x_{23}=D_2$
$, x_{24}=D_2, x_{25}=D_2, x_{26}=D_2, x_{01}=140-0.6D_1-D_2, x_{02}=140-0.5D_1-D_2,$
$x_{03}=140-0.2D_1-D_2, x_{04}=140-3D_1-D_2, x_{05}=140-1.5D_1-D_2, x_{06}=140-0.2D_1-D_2$ .

Note that all other *y* and *z* variables are zero.

$C(D) = [0.(x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26}) +$
$6.(y_{11} + y_{12} + y_{13} + y_{14} + y_{15}) + 2.5.(y_{21} + y_{22} + y_{23} + y_{24} + y_{25}) +$
$8.(z_{12} + z_{13} + z_{14} + z_{15} + z_{16}) + 4.(z_{22} + z_{23} + z_{24} + z_{25} + z_{26})] = 0$.

The NLP structure is:

$$\sum_{j=1}^{2} P_j(D_j).D_j \sum_{t=1}^{6} \beta_{jt} - C(D) = (30-0.2D_1).D_1.6 + (30-0.2D_2).D_2.6 - 0$$
$$= -1.2D_1^2 - 1.2D_2^2 + 180D_1 + 180D_2.$$

$$A_1 = \begin{Bmatrix} t=1; & x_{01} \in B(D) & \rightarrow & 0.6D_1 + D_2 \le 140 \\ t=2; & x_{02} \in B(D) & \rightarrow & 0.5D_1 + D_2 \le 140 \\ t=3; & x_{03} \in B(D) & \rightarrow & 0.2D_1 + D_2 \le 140 \\ t=4; & x_{04} \in B(D) & \rightarrow & 3D_1 + D_2 \le 140 \\ t=5; & x_{05} \in B(D) & \rightarrow & 1.5D_1 + D_2 \le 140 \\ t=6; & x_{06} \in B(D) & \rightarrow & 0.2D_1 + D_2 \le 140 \end{Bmatrix}$$

**2.**

$A_2, A_3 = \phi$
$\Omega'(B(D)) = A_1$.

**3.**

$$Max_{D_1,D_2 \ge 0}\{-1.2D_1^2 - 1.2D_2^2 + 180D_1 + 180D_2\}$$
*subject to*:
$t=1;\ 0.6D_1 + D_2 \le 140$
$t=2;\ 0.5D_1 + D_2 \le 140$
$t=3;\ 0.2D_1 + D_2 \le 140$
$t=4;\ 3D_1 + D_2 \le 140$
$t=5;\ 1.5D_1 + D_2 \le 140$
$t=6;\ 0.2D_1 + D_2 \le 140$ *and*
$D_1 + D_2 \le 140$.

*Optimal Solution*: $D_1^* = 27\ ,\ D_2^* = 59\ ,\ \pi(D_1^*, D_2^*) = 10428$

**4.** $V = \{(B(D), 10428)\}$.

Note that the corresponding objective value is given to each element in set *U.*

**5.** $t=4;\ 3D_1 + D_2 \le 140$ Subject to $D_1^* = 27\ ,\ D_2^* = 59$ is binding. *i=1.*

**6.** Since the binding constraint belongs to set $A_1$, the candidate bases for it are:

$B_1(D) = B(D) - \{x_{04}\} + \{y_{23}\} =$
$\{x_{01}, x_{02}, x_{03}, y_{23}, x_{05}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}\}$
$B_2(D) = B(D) - \{x_{04}\} + \{z_{25}\} =$
$\{x_{01}, x_{02}, x_{03}, z_{25}, x_{05}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}\}$.

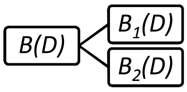



*Figure 3.4. Labelling the found candidate bases for the binding constraint*

**7.** $U_1 = \{B_1(D), B_2(D)\}$.

**8.** Suppose that the algorithm chooses *$B_1(D)$* at random.

***8.1.*** $B'(D) = B_1(D) =$

$\{x_{01}, x_{02}, x_{03}, y_{23}, x_{05}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}\}$.

***8.2.*** $x_{11}=0.6D_1, x_{12}=0.5D_1, x_{13}=0.2D_1, x_{14}=3D_1, x_{15}=1.5D_1, x_{16}=0.2D_1, x_{21}=D_2, x_{22}=D_2,$
$x_{23}=2D_2+3D_1-140, x_{24}=140-3D_1, x_{25}=D_2, x_{26}=D_2, x_{01}=140-0.6D_1-D_2,$
$x_{02}=140-0.5D_1-D_2, x_{03}=280-3.2D_1-2D_2, y_{23}=3D_1+D_2-140,$
$x_{05}=140-1.5D_1-D_2, x_{06}=140-0.2D_1-D_2$.

Note that *$x_{04}$* is a non-basic, and *$y_{23}$* is a basic variable.

$$C(D) = [0.(x_{11}+x_{12}+x_{13}+x_{14}+x_{15}+x_{16}+x_{21}+x_{22}+x_{23}+x_{24}+x_{25}+x_{26})+ \\ 6.(y_{11}+y_{12}+y_{13}+y_{14}+y_{15})+2.5.(y_{21}+y_{22}+y_{23}+y_{24}+y_{25})+ \\ 8.(z_{12}+z_{13}+z_{14}+z_{15}+z_{16})+4.(z_{22}+z_{23}+z_{24}+z_{25}+z_{26})] = 7.5D_1+2.5D_2-350.$$

The NLP structure is:

$$(30-0.2D_1).D_1.6+(30-0.2D_2).D_2.6-7.5D_1-2.5D_2+350 \\ = -1.2D_1^2-1.2D_2^2+172.5D_1+177.5D_2+350.$$

$$A_1 = \begin{Bmatrix} t=1; & x_{01}\in B(D) & \rightarrow & 0.6D_1+D_2 \le 140 \\ t=2; & x_{02}\in B(D) & \rightarrow & 0.5D_1+D_2 \le 140 \\ t=3; & x_{03}\in B(D) & \rightarrow & 3.2D_1+2D_2 \le 280 \\ t=5; & x_{05}\in B(D) & \rightarrow & 1.5D_1+D_2 \le 140 \\ t=6; & x_{06}\in B(D) & \rightarrow & 0.2D_1+D_2 \le 140 \end{Bmatrix}$$

$$A_2 = \{t=4; \quad x_{04}\notin B(D) \quad \rightarrow \quad 3D_1+D_2 > 140\}$$

$$A_3 = \{t=4; \quad x_{04}\notin B(D) \quad \rightarrow \quad 3D_1 \le 140\}$$

$\Omega'(B(D)) = A_1 \cup A_2 \cup A_3$.

***8.3.*** *$\Omega'(B(D))$* has feasible area, solve the *NLP*:

$$Max_{D_1,D_2\ge 0}\{-1.2D_1^2-1.2D_2^2+172.5D_1+177.5D_2+350\}$$
$$subject\ to:$$
$$t=1;\quad 0.6D_1+D_2\le 140$$
$$t=2;\quad 0.5D_1+D_2\le 140$$
$$t=3;\quad 3.2D_1+2D_2\le 280$$
$$t=4;\quad 3D_1+D_2>140$$
$$t=4;\quad 3D_1\le 140$$
$$t=5;\quad 1.5D_1+D_2\le 140$$
$$t=6;\quad 0.2D_1+D_2\le 140\qquad and$$
$$D_1+D_2\le 140.$$

$$Optimal\ Solution:\quad D_1^*=46.67\ ,\ D_2^*=65.33\ ,\ \pi(D_1^*,D_2^*)=12261.20$$

***8.4.*** $V=\{(B(D),10428),(B_1(D),12261.20)\}$ and $U_1=\{B_2(D)\}$.

***8.5.*** $t=3;\quad 3.2D_1+2D_2\le 280$ from set $A_1$ and $t=4;\quad 3D_1\le 140$ from set $A_3$ subject to $D_1^*=46.67\ ,\ D_2^*=65.33$ are binding. *i=2.*

***8.6.*** For the binding constraint which belongs to set $A_1$ , the candidate basis are defined as follow:

$$B_{11}(D)=B'(D)-\{x_{03}\}+\{y_{22}\}$$
$$\{x_{01},x_{02},y_{22},y_{23},x_{05},x_{06},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26}\}.$$

$$t=3\ :\ \theta(3,1):Min\{t'\ge 3;\ y_{1t'}\notin B(D)\}=3$$
$$\theta(3,2):Min\{t'\ge 3;\ y_{2t'}\notin B(D)\}=4$$
$$\tau'(3):Max\{\theta(3,j);\ j=1,2\}=4$$
$$b(3):\{j:\ \theta(3,j)=\tau'(3)\ \ and\ \ s_j-h_j\le s_i-h_i\ \ for\ i\ne j\}=2.$$

$$B_{12}(D)=B'(D)-\{y_{23}\}+\{z_{25}\}$$
$$\{x_{01},x_{02},x_{03},z_{25},x_{05},x_{06},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26}\}.$$

For the binding constraint which belongs to set $A_3$, the candidate bases are:

$$t=4\ :\ i(4)=2$$

$$B_{13}(D)=B'(D)-\{x_{24}\}+\{y_{13}\}$$
$$\{x_{01},x_{02},x_{03},y_{23},x_{05},x_{06},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},x_{21},x_{22},x_{23},y_{13},x_{25},x_{26}\}$$
$$B_{14}(D)=B'(D)-\{x_{24}\}+\{z_{15}\}$$
$$\{x_{01},x_{02},x_{03},y_{23},x_{05},x_{06},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},x_{21},x_{22},x_{23},z_{15},x_{25},x_{26}\}.$$

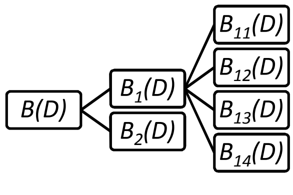



*Figure 3.5. Labelling the found candidate bases for the binding constraint*

Due to the limitations in bringing the long computational details in the thesis, we summarise the result of the remaining steps in Table 3.2. Continuing in this way finds the following candidate bases which have been visited and shown.

*Table 3.2. List of visited basis stemmed from the continued algorithm.*

| **The Visited Basis** | **Elements of the Basis** |
|---|---|
| $B_2(D)$ | $\{x_{01}, x_{02}, x_{03}, z_{25}, x_{05}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}\}$ |
| $B_{11}(D)$ | $\{x_{01}, x_{02}, y_{22}, y_{23}, x_{05}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}\}$ |
| $B_{13}(D)$ | $\{x_{01}, x_{02}, x_{03}, y_{23}, x_{05}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, y_{13}, x_{25}, x_{26}\}$ |
| $B_{14}(D)$ | $\{x_{01}, x_{02}, x_{03}, y_{23}, x_{05}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, z_{15}, x_{25}, x_{26}\}$ |
| $B_{22}(D)$ | $\{x_{01}, x_{02}, x_{03}, z_{25}, z_{26}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}\}$ |
| $B_{142}(D)$ | $\{x_{01}, x_{02}, x_{03}, z_{25}, x_{05}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, z_{15}, x_{25}, x_{26}\}$ |
| $B_{132}(D)$ | $\{x_{01}, x_{02}, x_{03}, z_{25}, x_{05}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, y_{13}, x_{25}, x_{26}\}$ |
| $B_{223}(D)$ | $\{x_{01}, x_{02}, x_{03}, y_{23}, z_{26}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}\}$ |
| $B_{113}(D)$ | $\{x_{01}, x_{02}, y_{22}, y_{23}, x_{05}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, y_{13}, x_{25}, x_{26}\}$ |
| $B_{144}(D)$ | $\{x_{01}, x_{02}, x_{03}, y_{23}, z_{26}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, z_{15}, x_{25}, x_{26}\}$ |
| $B_{112}(D)$ | $\{x_{01}, x_{02}, y_{22}, y_{23}, z_{26}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}\}$ |
| $B_{222}(D)$ | $\{x_{01}, x_{02}, x_{03}, z_{25}, z_{26}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, z_{15}, x_{25}, x_{26}\}$ |

| | |
|---|---|
| $B_{114}(D)$ | $\{x_{01}, x_{02}, y_{22}, y_{23}, x_{05}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, z_{15}, x_{25}, x_{26}\}$ |
| $B_{221}(D)$ | $\{x_{01}, x_{02}, x_{03}, z_{25}, z_{26}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, y_{13}, x_{25}, x_{26}\}$ |
| $B_{111}(D)$ | $\{x_{01}, x_{02}, y_{22}, y_{23}, y_{24}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}\}$ |
| $B_{1132}(D)$ | $\{x_{01}, x_{02}, y_{22}, y_{23}, z_{26}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, y_{13}, x_{25}, x_{26}\}$ |
| $B_{2211}(D)$ | $\{x_{01}, x_{02}, x_{03}, y_{23}, z_{26}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, y_{13}, x_{25}, x_{26}\}$ |
| $B_{1131}(D)$ | $\{x_{01}, x_{02}, y_{22}, y_{23}, y_{24}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, y_{13}, x_{25}, x_{26}\}$ |
| $B_{1441}(D)$ | $\{x_{01}, x_{02}, y_{22}, y_{23}, z_{26}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, z_{15}, x_{25}, x_{26}\}$ |
| $B_{11321}(D)$ | $\{x_{01}, y_{21}, y_{22}, y_{23}, z_{26}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, y_{13}, x_{25}, x_{26}\}$ |

Here we detail some of the final steps of the algorithm:

**1.** This step is in fact the application of the step 9 and its sub steps. Suppose that the algorithm chooses $B_{11311}(D)$ at random.

**1.1.** $B^{'}(D) = B_{11311}(D)$
$= \{x_{01}, y_{21}, y_{22}, y_{23}, y_{24}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, y_{13}, x_{25}, x_{26}\}.$

**1.2.**
$$\begin{aligned}
&x_{11}= 0.6D_1, x_{12}= 0.5D_1, x_{13}= 3.2D_1 - 140, x_{14}= 140, x_{15}= 1.5D_1, x_{16}= 0.2D_1,\\
&x_{21}= 5.2D_1 + 5D_2 - 560, x_{22}= 140 - 0.5D_1, x_{23}= 280 - 3.2D_1, y_{13}= 3D_1 - 140,\\
&x_{25}= 140 - 1.5D_1, x_{26}= D_2, x_{01}= 700 - 5.8D_1 - 5D_2,\\
&y_{21}= 5.2D_1 + 4D_2 - 560, y_{22}= 4.7D_1 + 3D_2 - 420,\\
&y_{23}= D_2, y_{24}= 1.5D_1 + D_2 - 140, x_{06}= 140 - 0.2D_1 - D_2.
\end{aligned}$$

Note that $x_{05}$, $x_{04}$, $x_{03}$, $x_{02}$ and $x_{24}$ are non-basic, and $y_{24}$, $y_{23}$, $y_{22}$, $y_{21}$ and $y_{13}$ are basic variables.

$$\begin{aligned}
C(D) = [&0.(x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26}) +\\
&6.(y_{11} + y_{12} + y_{13} + y_{14} + y_{15}) + 2.5.(y_{21} + y_{22} + y_{23} + y_{24} + y_{25}) +\\
&8.(z_{12} + z_{13} + z_{14} + z_{15} + z_{16}) + 4.(z_{22} + z_{23} + z_{24} + z_{25} + z_{26})] = 50.25D_1 + 25D_2 - 3990.
\end{aligned}$$

The NLP structure is:

$(30-0.2D_1).D_1.6+(30-0.2D_2).D_2.6-50.25D_1-25D_2+3990$
$=-1.2D_1^2-1.2D_2^2+129.75D_1+155D_2+3990$

$$A_1=\begin{Bmatrix} t=1; & x_{01}\in B(D) & \rightarrow & 5.8D_1+5D_2\le 700 \\ t=6; & x_{06}\in B(D) & \rightarrow & 0.2D_1+D_2\le 140 \end{Bmatrix}$$

$$A_2=\begin{Bmatrix} t=2; & x_{02}\notin B(D) & \rightarrow & 5.2D_1+4D_2>560 \\ t=3; & x_{03}\notin B(D) & \rightarrow & 4.7D_1+3D_2>420 \\ t=4; & x_{04}\notin B(D) & \rightarrow & 3D_1>140 \\ t=5; & x_{05}\notin B(D) & \rightarrow & 1.5D_1+D_2>140 \end{Bmatrix}$$

$$A_3=\begin{Bmatrix} t=2; & x_{02}\notin B(D) & \rightarrow & 0.5D_1\le 140 \\ t=3; & x_{03}\notin B(D) & \rightarrow & 3.2D_1\le 280 \\ t=4; & x_{04}\notin B(D) & \rightarrow & 140>0 \\ t=5; & x_{05}\notin B(D) & \rightarrow & 1.5D_1\le 140 \end{Bmatrix}$$

$\Omega'(B(D))=A_1\cup A_2\cup A_3$ .

**1.3.** *Ω'(B(D))* has feasible area, solve the *NLP*:

$Max_{D_1,D_2\ge 0}\{-1.2D_1^2-1.2D_2^2+138.75D_1+161D_2+3150\}$
*subject to* :
$t=1;\quad x_{01}\in B(D)\quad \rightarrow\quad 5.8D_1+5D_2\le 700$
$t=2;\quad 5.2D_1+4D_2>560$
$t=2;\quad 0.5D_1\le 140$
$t=3;\quad 4.7D_1+3D_2>420$
$t=3;\quad 3.2D_1\le 280$
$t=4;\quad 3D_1>140$
$t=5;\quad 1.5D_1+D_2>140$
$t=5;\quad 1.5D_1\le 140$
$t=6;\quad 0.2D_1+D_2\le 140\quad and$
$D_1+D_2\le 140$ .

*Optimal Solution* : $D_1^*=56.54$ , $D_2^*=66.49$ , $\pi(D_1^*,D_2^*)=12490.748$

**1.4.** $$V = \left\{ \begin{array}{l} (B(D),10428),(B_1(D),12261.20), \\ (B_2(D),11260.21),(B_{11}(D),12308.33), \\ (B_{13}(D),12277.9),(B_{14}(D),\ 11652.08), \\ (B_{22}(D),11935.05),(B_{142}(D),10476.67), \\ (B_{132}(D),10476.67),(B_{223}(D),20339.38), \\ (B_{113}(D),12387.98),(B_{144}(D),12401.97), \\ (B_{112}(D),12308.39),(B_{222}(D),11947.39), \\ (B_{114}(D),12313.63),(B_{221}(D),12184.46), \\ (B_{111}(D),12308.33),(B_{1132}(D),12559.71), \\ (B_{2211}(D),0),(B_{1131}(D),11790.748), \\ (B_{1441}(D),12443.85),(B_{11321}(D),12559.71), \\ (B_{11311}(D),12490.75) \end{array} \right\}$$ and

$U_5 = \{B_{11322}(D)\}$.

**1.5.** $t = 2;\ \ 5.2D_1 + 4D_2 > 560$ from set $A_2$ subject to $D_1^* = 56.54$ , $D_2^* = 66.49$ is binding. *i=6.*

**1.6.** For the binding constraint which belongs to set $A_2$ , the candidate basis are defined as follow:

$B_{113111}(D) = B'(D) - \{y_{21}\} + \{x_{02}\}$

$\{x_{01}, x_{02}, y_{22}, y_{23}, y_{24}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, y_{13}, x_{25}, x_{26}\}$.

Candidate $B_{113111}(D)$ is like $B_{1131}(D)$ in set $V$ . So set $U_6$ is still not constructed, but matrix *POS* should be updated.

$$POS = \begin{pmatrix} (D_1^* = 46.67\ ,\ D_2^* = 65.33) & B_1(D) & \pi(D) = 12261.20 \\ (D_1^* = 48.64\ ,\ D_2^* = 62.17) & B_{13}(D) & \pi(D) = 12277.9 \\ (D_1^* = 36.29\ ,\ D_2^* = 58.35) & B_2(D) & \pi(D) = 11260.21 \\ (D_1^* = 46.67\ ,\ D_2^* = 66.89) & B_{22}(D) & \pi(D) = 11935.05 \\ (D_1^* = 56.70\ ,\ D_2^* = 64.33) & B_{144}(D) & \pi(D) = 12401.97 \\ (D_1^* = 51.19\ ,\ D_2^* = 63.22) & B_{113}(D) & \pi(D) = 12387.98 \\ (D_1^* = 59.2\ ,\ D_2^* = 59.78) & B_{221}(D) & \pi(D) = 12184.46 \\ (D_1^* = 58.6\ ,\ D_2^* = 67.72) & B_{1132}(D) & \pi(D) = 12559.71 \\ (D_1^* = 57.29\ ,\ D_2^* = 70.2) & B_{1441}(D) & \pi(D) = 12443.85 \\ (D_1^* = 56.54\ ,\ D_2^* = 66.49) & B_{1131}(D) & \pi(D) = 11790.75 \end{pmatrix}.$$

**2.** *i=5* and restart from step 9; continue till set $U_5$ comes to $\phi$ .

**3.** The only element in set $U_5$ is $B_{11322}(D)$. This step is again repeating the step 9 of the algorithm.

**3.1.** $B'(D)=B_{11322}(D)=$

$\{x_{01}, x_{02}, y_{22}, z_{25}, z_{26}, x_{06}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, y_{13}, x_{25}, x_{26}\}$.

**3.2.**
$$x_{11} = 0.6D_1, x_{12} = 0.5D_1, x_{13} = 3.2D_1 - 140, x_{14} = 140, x_{15} = 1.5D_1, x_{16} = 0.2D_1,$$
$$x_{21} = D_2, x_{22} = 3.2D_1 + 2D_2 - 140, x_{23} = 140 - 3.2D_1, y_{13} = 3D_1 - 140,$$
$$x_{25} = 140 - 1.5D_1, x_{26} = 1.5D_1 + 3D_2 - 140, x_{01} = 140 - 0.6D_1 - D_2,$$
$$x_{02} = 420 - 3.7D_1 - 2D_2, y_{22} = 3.2D_1 + D_2 - 140,$$
$$z_{25} = D_2, z_{26} = 1.5D_1 + 2D_2 - 140, x_{06} = 280 - 1.7D_1 - 3D_2 .$$

Note that $x_{05}$ , $x_{04}$ , $x_{03}$ and $x_{24}$ are non-basic, and $z_{26}$ , $z_{25}$ , $y_{22}$ and $y_{13}$ are basic variables.

$$C(D) = [0.(x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26}) +$$
$$6.(y_{11} + y_{12} + y_{13} + y_{14} + y_{15}) + 2.5.(y_{21} + y_{22} + y_{23} + y_{24} + y_{25}) +$$
$$8.(z_{12} + z_{13} + z_{14} + z_{15} + z_{16}) + 4.(z_{22} + z_{23} + z_{24} + z_{25} + z_{26})] = 32D_1 + 14.5D_2 - 1470.$$

The NLP structure is:

$$(30 - 0.2D_1).D_1.6 + (30 - 0.2D_2).D_2.6 - 32D_1 - 14.5D_2 + 1470$$
$$= -1.2D_1^2 - 1.2D_2^2 + 148D_1 + 165.5D_2 + 1470$$

$$A_1 = \begin{Bmatrix} t = 1; & x_{01} \in B(D) & \rightarrow & 0.6D_1 + D_2 \le 140 \\ t = 2; & x_{02} \in B(D) & \rightarrow & 3.7D_1 + 2D_2 \le 420 \\ t = 6; & x_{06} \in B(D) & \rightarrow & 1.7D_1 + 3D_2 \le 280 \end{Bmatrix}$$

$$A_2 = \begin{Bmatrix} t = 3; & x_{03} \notin B(D) & \rightarrow & 3.2D_1 + D_2 > 280 \\ t = 4; & x_{04} \notin B(D) & \rightarrow & 3D_1 > 140 \\ t = 5; & x_{05} \notin B(D) & \rightarrow & 1.5D_1 + 2D_2 > 140 \end{Bmatrix}$$

$$A_3 = \begin{Bmatrix} t = 3; & x_{03} \notin B(D) & \rightarrow & 3.2D_1 \le 280 \\ t = 4; & x_{04} \notin B(D) & \rightarrow & 140 > 0 \\ t = 5; & x_{05} \notin B(D) & \rightarrow & 1.5D_1 \le 140 \end{Bmatrix}$$

$$\Omega'(B(D)) = A_1 \cup A_2 \cup A_3 .$$

**3.3.** *Ω'(B(D))* has feasible area, solve the *NLP*:

$$Max_{D_1,D_2\geq 0}\{-1.2D_1^2-1.2D_2^2+148D_1+165.5D_2+1470\}$$

*subject to*:

$t=1;\quad 0.6D_1+D_2\leq 140$

$t=2;\quad 3.7D_1+2D_2\leq 420$

$t=3;\quad 3.2D_1+D_2>280$

$t=3;\quad 3.2D_1\leq 280$

$t=4;\quad 3D_1>140$

$t=5;\quad 1.5D_1+2D_2>140$

$t=5;\quad 1.5D_1\leq 140$

$t=6;\quad 1.7D_1+3D_2\leq 280$

$D_1+D_2\leq 140.$

$Optimal\ \ Solution:\ D_1^*=70.89\ ,\ D_2^*=53.16\ \ ,\ \pi(D_1^*,D_2^*)=11338.30$

**3.4.**

$$V=\left\{\begin{array}{l}(B(D),10428),(B_1(D),12261.20),\\ (B_2(D),11260.21),(B_{11}(D),12308.33),\\ (B_{13}(D),12277.9),(B_{14}(D),11652.08),\\ (B_{22}(D),11935.05),(B_{142}(D),10476.67),\\ (B_{132}(D),10476.67),(B_{223}(D),20339.38),\\ (B_{113}(D),12387.98),(B_{144}(D),12401.97),\\ (B_{112}(D),12308.39),(B_{222}(D),11947.39),\\ (B_{114}(D),12313.63),(B_{221}(D),12184.46),\\ (B_{111}(D),12308.33),(B_{1132}(D),12559.71),\\ (B_{2211}(D),0),(B_{1131}(D),11790.748),\\ (B_{1441}(D),12443.85),(B_{11321}(D),12559.71),\\ (B_{11311}(D),12490.75),(B_{11322}(D),11338.30)\end{array}\right\}$$ and

$U_5=\phi.$

**3.5.** $t=3;\quad 3.2D_1+D_2>280$ from set $A_2$ and $t=6;\quad 1.7D_1+3D_2\leq 280$ from set $A_1$ subject to $D_1^*=70.89\ ,\ D_2^*=53.16$ are binding. *i=6.*

**3.6.** For the binding constraint which belongs to set $A_2$ , the candidate basis are defined as follow:

$B_{113221}(D)=B'(D)-\{y_{22}\}+\{x_{03}\}$

$\{x_{01},x_{02},x_{03},z_{25},z_{26},x_{06},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},x_{21},x_{22},x_{23},y_{13},x_{25},x_{26}\}$

$$\begin{array}{ll} t=6\ \ : & \alpha(6,1):Max\{t'\leq 6;z_{1t'}\notin B(D)\}=6\\ & \alpha(6,2):Max\{t'\leq 6;z_{2t'}\notin B(D)\}=4\\ & \tau(6):Min\{\alpha(6,j);j=1,2\}=4\\ & a(6):\{j:\ \ \alpha(6,j)=\tau(6)\ \ and\ \ s_j-h_j\geq s_i-h_i\ \ for\ i\neq j\}=2.\end{array}$$

$B_{113222}(D)=B'(D)-\{z_{25}\}+\{y_{23}\}$

$\{x_{01},x_{02},y_{22},y_{23},z_{26},x_{06},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},x_{21},x_{22},x_{23},y_{13},x_{25},x_{26}\}.$

**3.7.** Candidate $B_{113221}(D)$ is like $B_{221}(D)$ and candidate $B_{113222}(D)$ is like $B_{1132}(D)$ in matrix *POS* . So set $U_6$ is still not constructed.

**4.** $i=5$ and set $U_5$ is empty now.

**5.** $i=6$ and set $U_6$ has not been constructed. So the step 12 and further steps of the main algorithm should be executed.

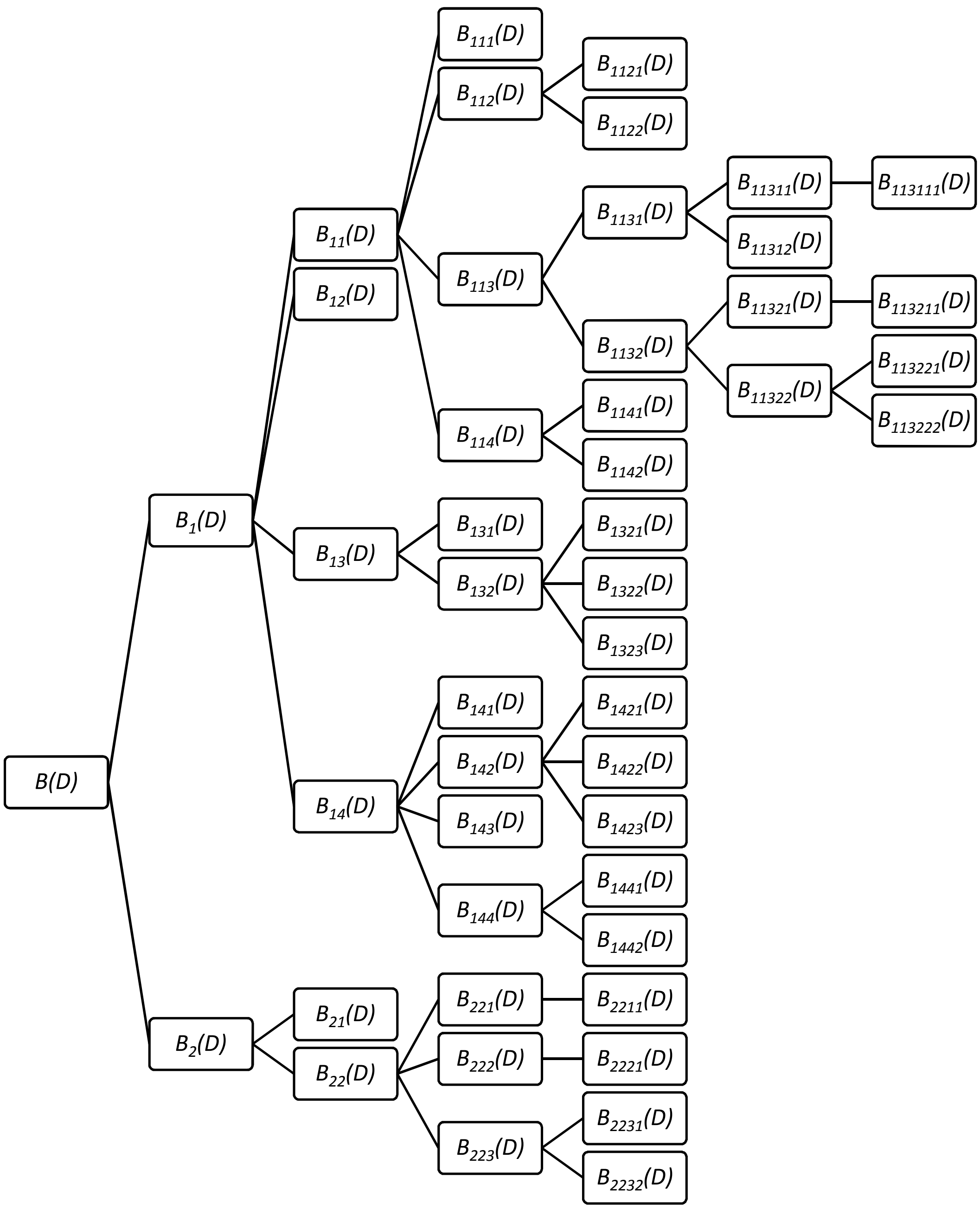



*Figure 3.6. The tree structure of the solutions of the example*

**6.** The biggest value of *π(D)* in *POS* and the corresponding $D^*$ and basis for that is as follow: $D_1^* = 58.6$ , $D_2^* = 67.72 \quad B_{1132}(D) \quad \pi(D) = 12559.71$

**7.** The optimal price for each product is:

$$p_1(D_1) = 30 - 0.2(D_1) = 18.28$$
$$p_2(D_2) = 30 - 0.2(D_2) = 16.45$$

The optimal values of x, y, z variables corresponding to the optimal basis are:

$x_{11} = 0.6D_1 = 0.6(58.6) = 35.16 \qquad x_{12} = 0.5D_1 = 0.5(58.6) = 29.3$

$x_{13} = 3.2D_1 - 140 = 3.2(58.6) - 140 = 47.52 \qquad x_{14} = 140$

$x_{15} = 1.5D_1 = 1.5(58.6) = 87.9 \qquad x_{16} = 0.2D_1 = 0.2(58.6) = 11.72$

$x_{21} = 67.72$

$x_{22} = 67.72$

$x_{23} = 280 - 3.2D_1 = 280 - 3.2(58.6) = 92.48$

$y_{13} = 3D_1 - 140 = 3(58.6) - 140 = 35.8$

$x_{25} = 140 - 1.5D_1 = 140 - 1.5(58.6) = 52.1$

$x_{26} = 1.5D_1 + 2D_2 - 140 = 1.5(58.6) + 2(67.72) - 140 = 83.34$

$y_{22} = 3.2D_1 + 2D_2 - 280 = 3.2(58.6) + 2(67.72) - 280 = 42.96$

$y_{23} = 67.72$

$z_{26} = 1.5D_1 + D_2 - 140 = 1.5(58.6) + 67.72 - 140 = 15.62$

## 3.4. Conclusion

In this Chapter we have presented a mathematical programming model for determining the optimal production and constant pricing policy for a finite time horizon multiproduct production system with capacity constraints. Our model allows for backorders. Demand for each product is deterministic and dependent on its price, and the production set up cost is negligible. The solution strategy to approach the specified problem is an iterative two stage algorithm. The algorithm solves the nonlinear programming problem only under linear constraints, although keeps the nonlinear constraints feasibility. The first stage finds the value of a Linear Programming's objective function in terms of the main problems' decision variables and in the second stage a Non-Linear Programming problem is solved subject to a linear constraint set. We illustrate the method with a detailed numerical example. The result of this Chapter has been published by Caccetta and Mardaneh (2010).

# Chapter 4

# Optimal Dynamic Pricing and Production Planning for Multiproduct Multiperiod Systems with Backorders

The purpose of this Chapter is to develop a mathematical programming model which optimises the production planning and pricing of a manufacturing system with multiple products over a multiple period horizon to maximize the total profit which consists of sales revenue, production, inventory holding and backordering costs. The application that motivated this research is manufacturing pricing, where the products are non-perishable assets and can be stored to fulfil the future demands. We assume that the firm does not change the price list very frequently. However, the developed model and its solution strategy have the capability to handle the general case of manufacturing systems with frequent time-varying price lists. We present an alternative model to the formulation presented in Section 2.3. The new model is called "Two-Stage Optimisation Model". Due to the complexity of the problem formulated in Chapter 2, in this Chapter we reformulate the problem in a way that can be handled with an efficient iterative solution strategy. We extend the previous Chapter's algorithm for constant pricing to the time-variant pricing problem, which is the aim of this project as discussed in earlier Chapters.

Our problem is computationally difficult, because it involves a nonlinear objective function and some nonlinear constraints. Our strategy to reduce the level of difficulty in the "Two-Stage Optimisation Model" is to utilise a method that solves a relaxed problem with only linear constraints. However, our method keeps track of the feasibility with respect to the nonlinear constraints in the original problem. The developed model is a combination of Linear Programming (LP) and linearly constrained Nonlinear Programming (NLP) which is solved iteratively. Iterations consist of two main stages: The first stage starts with a given basis for the LP, solves the linear equations corresponding to the chosen basis and finds the value of the LP's objective function in terms of the main problems' decision variables (which is the demand intensity of each product induced by the pricing policy). The second stage receives the output of the first stage and based on that, finds the structure of the NLP's objective function. Then, the updated linear constraint set corresponding to the chosen basis in stage 1 is defined and the NLP is solved subject to the determined updated linear constraint set. Depending on the result of the NLP solution, some candidate bases will be revealed to restart iteration and repeat stages 1 and 2. To achieve the final optimal solution, a branching type procedure is utilised which will stop given that all next level branches have been visited in earlier iterations. Bearing in mind the fact that the backorder case makes the problem fairly difficult to solve, our proposed strategy is practical to implement, as we demonstrate through a numerical example.

This Chapter is organised as follows. In Section 4.1 we present the problem features and formulate it in a Two Stage Optimisation Model. Section 4.2 explains a solution strategy for the developed model and detailed algorithms along with implementation are presented in Section 4.3 through a numerical example.

## 4.1. A Two-Stage Optimisation Model

This Section formulates the problem of coordinated pricing and production planning of multiple products with allowable inventory carrying and backorders. The following features of the problem, which we model, are also listed in Chapter 2.

- The firm produces *n* different products and the demand of each product is period-varying and seasonal over the planning horizon.
- Demand of each product is deterministic and dependent on its price.
- The production capacity is limited and shared among different products.
- Products use the same amount of capacity; here each product uses one unit of capacity.
- The production set up cost is negligible.

However there are some additional features which are:

- The planning horizon consists of m blocks of T periods.
- The price of each product is constant over each block but is varying in different blocks.
- All related costs including production, holding and shortage costs are constant over each block but are varying in different blocks.
- Demand of each product is period-varying and seasonal over the planning horizon.

We make use of the following notation and terminology in our models.

**Parameters:**

*m :* the number of time blocks

*T :* the number of periods in each block

$b$ : the block number, $b = 1,2,\ldots,m$. Note that block $b$ includes time periods $T(b-1)+1$ , $T(b-1)+2$ , … , $T(b-1)+T$.

$c_{jb}$ : the production cost of one unit of item $j$ in time block $b$; $j=1,2,\ldots,n$ and $b= 1,2,\ldots,m$

$h_{jb}$ : the holding cost of one unit of item $j$ in inventory for one period in time block $b$

$s_{jb}$ : the backordering cost of one unit of item $j$ for one period in time block $b$

$\beta_{jt}$ : the seasonality parameter of item $j$ in period $t$

**Variables:**

$p_{jb}$ : the price of product $j$ in time block $b$; $j=1,2,\ldots,n$ and $b= 1,2,\ldots,m$

$p_b$ : the n-dimensional price vector of time block $b$

$D_b$ : the n-dimensional demand intensity vector of time block $b$

$x_{jt}$ : the amount of product $j$ produced in period $t$.

$y_{jt}$ : the amount of product $j$ held in inventory at the end of period $t$.

$z_{jt}$ : the amount of product $j$ backordered from period $t$ to meet the demand of period $t-1$.

$D_{jt}$ : the demand for item $j$ in period $t$; $j=1,2,\ldots,n$ and $t=1,2,\ldots,mT$ as $D_{jt}= \beta_{jt} D_{jb}$.

$x_{0t}$ : the amount of unused capacity in period $t$

$X$ : the $n \times mT$ production matrix

$Y$ : the $n \times mT$ inventory matrix

$Z$ : the $n \times mT$ backordering matrix

$D$ : the $n \times m$ demand intensity matrix

**Functions:**

$D_{jb}(p)$ : the demand intensity for product $j$ in time block $b$, which is a function of price vector

Note that the relationship between demand intensity and price is known, but both of them are decision variables of the problem.

$R_{jb}(D)$ : the revenue function as $D_{jb}(p).\ p_{jb}$

$C(D)$ : the minimum cost of satisfying the demand corresponding to the induced demand intensities $D_{11}, ..., D_{n1}, D_{12}, ..., D_{n2}, ..., D_{1m}, ..., D_{nm}$

As before, we assume that corresponding to each demand intensity vector of each time block, there is just one price vector of the same time block. By this assumption, we have the ease of using $D_{jb}$ for $j = 1, \ldots, n$ and $b = 1, 2, \ldots, m$ as the decision variables. Also, the interaction between the demand of a product in one time block and the price of the product in other time blocks is negligible. In another words, there is no strategic behaviour of customers in this model. The next assumption relies on the concavity of the revenue function, $R_j(D)$, for each product $j = 1, \ldots, n$. The seasonality model is assumed to be purely multiplicative: $D_{jt} = \beta_{jt} \cdot D_{jb}(p)$. We can justify this assumption by considering that the distribution of price sensitivity among the participants in the market doesn't change although the size of the market may differ in different periods. This justification is an interpretation of the model used for a single product in Gilbert (1999). As noted earlier, for the single product model, Kunreuther and Schrage (1973) assume a price-insensitive additive seasonality term with the intention that demand in period $t$ is expressed as $d_t(p) = \alpha_t + \beta_t D(p)$. Although this is more general than the purely multiplicative model, we note that in the application of their model, Kunreuther and Schrage assume that seasonality is purely additive, i.e., $\beta_1 = \beta_2 = \ldots \beta_T$. A further assumption we make is that the products are indexed in decreasing order of their holding and shortage costs, i.e., $h_i \geq h_j$ and $s_i \geq s_j$ for $i<j$. Like any other inventory system, the shortage cost is always more than holding cost.

The problem of jointly determining the price and production plan can be formally expressed as follows:

$$\pi = Max_{D_b \geq 0} \left\{ \pi(D) = \sum_{b=1}^{b=m} \sum_{j=1}^{j=n} R_{jb}(D) \sum_{t=bT-T+1}^{t=bT} \beta_{jt} - C(D) \right\} \qquad (4.1)$$

such that

$$\sum_{b=1}^{b=m}\sum_{j=1}^{j=n}\sum_{t=bT-T+1}^{t=bT} D_{jb}\beta_{jt} \leq \sum_{t=1}^{mT} K_t \,. \tag{4.2}$$

In (4.1),

$$C(D) = Min_{X,Y,Z\geq 0}\left\{\sum_{b=1}^{b=m}\sum_{j=1}^{j=n}\sum_{t=bT-T+1}^{t=bT}(c_{jb}x_{jt} + h_{jb}y_{jt} + s_{jb}z_{jt})\right\} \tag{4.3}$$

subject to:

$$D_{jt} = x_{jt} + y_{jt-1} + z_{jt+1} - y_{jt} - z_{jt}\ , for\ t = 1, \dots, mT\ and\ j = 1, \dots, n, \tag{4.4}$$

$$\sum_{j=0}^{n} x_{jt} = K_t\ , for\ t = 1, \dots, mT \tag{4.5}$$

$$y_{it}z_{jt+1} = 0\ , for\ t = 1, \dots, mT\ and\ i, j = 1, \dots, n \tag{4.6}$$

$$x_{jt}, y_{jt}, z_{jt}, D_{jt} \geq 0, for\ t = 1, \dots, mT\ and\ j = 0, \dots, n. \tag{4.7}$$

We refer to the problem (4.3)-(4.7) as the Cost Minimization Sub Problem, “CMSP”.

The objective function (4.1) consists of the sales revenue minus the total cost associated with the chosen demand intensity. Constraint (4.2) ensures that only demand intensity vectors which result a feasible solution to the CMSP have been considered. Constraints (4.4) are a set of flow balance equations that ensure that all of the induced demand is satisfied. Constraints (4.5) ensure that there is an adequate amount of capacity in period *t* to produce all *n* items based on the plan. The requirement in (4.6) that inventory and shortage as a cross product should be zero ensures that when there is an insufficient amount of capacity in one period from *t =1, . . . ,mT* the priority is to meet the demand of the same period instead of the others’ periods. Finally (4.7) is just the non-negativity series of constraints.

For the case when backorders are not allowed, Gilbert (2000) utilised the fact that the objective function to be maximized in the model is concave in the demand vector $D = [D_1, \ldots, D_n]$. Gilbert's method of proof can be used to establish that this property holds also for the backorder case in our model. Formally we have:

THEOREM 2. The profit function, $\pi(D)$ that is to be maximized in (4.1) is concave in the demand matrix.

PROOF. Take two demand matrices $D^1$ and $D^2$, and let $D^3 = \alpha D^1 + (1-\alpha) D^2$ where $\alpha \in (0, 1)$. Let $(x^1, y^1, z^1)$ and $(x^2, y^2, z^2)$ be the optimal solution variables associated with $C(D^1)$ and $C(D^2)$ respectively. Thus, $(x^1, y^1, z^1)$ and $(x^2, y^2, z^2)$ are feasible with respect to Constraints (4.4-4.7) when $D^1$ and $D^2$ are in the left-hand side of (4.4) respectively. Let $\tilde{x} = \alpha\, x^1 + (1-\alpha)x^2$, $\tilde{y} = \alpha\, y^1 + (1-\alpha)y^2$ and $\tilde{z} = \alpha\, z^1 + (1-\alpha)\, z^2$. Clearly, $(\tilde{x}, \tilde{y}, \tilde{z})$ satisfies all of the constraints in (4.4-4.7). Therefore:

$$\begin{aligned}
&\alpha\pi(D^1) + (1-\alpha)\pi(D^2) \\
&= \sum_{j=1}^{n}\sum_{b=1}^{m}(\alpha R_{jb}(D^1) + (1-\alpha)R_{jb}(D^2))\sum_{t=bT-T+1}^{bT}\beta_{jt} - \alpha C(D^1; x^1, y^1, z^1) \\
&\quad +(1-\alpha)C(D^2; x^2, y^2, z^2) \\
&= \sum_{j=1}^{n}\sum_{b=1}^{m}(\alpha R_{jb}(D^1) + (1-\alpha)R_{jb}(D^2))\sum_{t=bT-T+1}^{bT}\beta_{jt} - C(D^3; \tilde{x}, \tilde{y}, \tilde{z}),
\end{aligned} \tag{4.8}$$

where the latter equality follows from the linearity of the objective function in (4.3) and the fact that $(\tilde{x}, \tilde{y}, \tilde{z})$ is a feasible solution to (4.4-4.7) when $D^3$ is in the left-hand side of (4.4). By assumption, $R_{jb}(D)$ is concave. Therefore, $\alpha\, R_{jb}(D^1) + (1-\alpha)\, R_{jb}(D^2) \leq R_{jb}(D^3)$ for each $j = 1, \ldots, n$ and $b = 1, \ldots, m$. By substituting for each term in the summation of the right-hand side of (4.8) , we have:

$$\begin{aligned}
\alpha\pi(D^1) + (1-\alpha)\pi(D^2) &\leq \sum_{j=1}^{n}\sum_{b=1}^{m} R_{jb}(D^3)\sum_{t=bT-T+1}^{bT}\beta_{jt} - C(D^3; \tilde{x}, \tilde{y}, \tilde{z}) \\
&= \pi(D^3)
\end{aligned} \tag{4.9}$$

where the latter inequality results from the principle of optimality and the definition of $C(D^3)$ as the optimal (minimum) cost of satisfying demand $D^3$. □

## 4.2. Solution Strategy

Given the capacity limitations and other parameters, the firm must decide upon production quantities, inventory and backorder levels for each item as well as a constant price at which it commits to sell the products over the total planning horizon.

Note that for each *D* vector, as the decision variable of the problem, there is an optimal solution to the CMSP. In other words, when the *D* vector is changed the coefficients of the profit function, $\pi(D)$, will also change. Consequently, the problem can be solved iteratively. Each iteration starts with a known basis for the CMSP and involves two stages:

**Stage 1: Solve the cost minimization sub problem, CMSP**

1) Consider a known basis, $B(D)$, for the CMSP.

   Each basis consists of some of the x,y and z variables for $j=0,1,\ldots,n$ and $t=1,2,\ldots,mT$.

2) Find the values of the basic variables for CMSP in terms of $D_{jb}$s.

   This step can be done by using equations (4.4) and (4.5).

3) Find the value of the objective function of CMSP, $C(D)$, in terms of $D_{jb}$s by using (4.3).

**Stage 2: Solve the main Non-Linear problem and update the basis**

1) Restructure the profit function, $\pi(D)$, subject to the above defined $C(D)$ function and using (4.1).

2) Determine a linear constraint set $\Omega(B(D))$ such that $B(D)$ is an optimal basis for any $D' \in \Omega(B(D))$.

Note that once *B(D)* is determined, *Ω(B(D))* can be fully specified by using constraint set generation procedure presented in Section 4.2.1.

Theorem 2 guarantees that an optimal solution for the concave objective function, *π(D)*, over a linear set of constraints, *Ω(B(D)),* can be obtained.

3) Solve the *NLP,* that maximizes *π(D)* subject to $D \in \Omega(B(D))$.

4) If any of the constraints are binding in the optimal solution to this *NLP*, determine how the basis for the CMSP should change. Find the candidate new basis.

The procedure to do step 4 is detailed in Section 4.2.2.

### 4.2.1. Constraint set, Ω(B(D)), generation procedure.

**Input:**

The *x,y* and *z* basic variables.

**Output:**

A set of linear constraints, $\Omega(B(D)) = A_1 U\ A_2 U\ A_3$, in terms of $D_{jb}$s.

**Steps:**

1. Define *i(t)* as the number of $x_{jt}$ variables in the basis for *j=0,…,n*.
2. For each period *t,* if $x_{0t} \in B(D)$, calculate the following constraint in terms of $D_{jb}$s.

   $$\sum_{j=1}^{n} x_{jt} \leq K_t. \tag{4.10}$$

3. Add constraint (4.10) in a set named $A_1$.
4. If $x_{0t} \notin B(D)$ and $y_{i(t)t-1} \in B(D)$ , calculate the following constraint in terms of $D_{jb}$s.

   $$y_{i(t)t-1} > 0 \tag{4.11a}$$

   $$x_{i(t)t} \geq 0. \tag{4.11b}$$

5. If $x_{0t} \notin B(D)$ and $y_{i(t)t-1} \notin B(D)$, calculate the following constraint in terms of $D_{jb}$s.

$z_{i(t)t+1} > 0$ (4.11a)

$x_{i(t)t} \geq 0.$ (4.11b)

6. Add (4.11a) constraint in a set named $A_2$ and (4.11b) in a set named $A_3$.
7. Define $\Omega(B(D))= A_1 U\ A_2 U\ A_3$.

### 4.2.2. Finding a new candidate basis procedure.

**Input:**

Solution of *NLP* and list of binding constraint in $\Omega(B(D)) = A_1 U\ A_2 U\ A_3$.

**Output:**

Candidates for a change of the current basis, *B(D)*.

Before going to the main steps of the procedure we need to define some values as follow:

$$\theta(t,j): Min\left\{t' \geq t;\ y_{jt'} \notin B(D)\right\}$$

$$\alpha(t,j): Max\left\{t' \leq t;\ z_{jt'} \notin B(D)\right\}$$

$$\tau(t): Min\ \{\alpha(t,j);\ j = 1, \ldots, n\}$$

$$a(t): \left\{j: \alpha(t,j) = \tau(t)\ and\ s_j - h_j \geq s_i - h_i\ for\ i \neq j\right\}$$

$$\tau'(t): Max\ \{\theta(t,j);\ j = 1, \ldots, n\}$$

$$b(t): \left\{j: \theta(t,j) = \tau'(t)\ and\ s_j - h_j \leq s_i - h_i\ for\ i \neq j\right\}.$$

**Steps:**

1. For any binding constraint from set $A_2$ , determine the new candidates as follows:
   - $If\ \left(z_{i(t)t+1} \in B(D), for\ j = 1, \ldots, n\right)\left\{first\ candidate\ basis = B(D) - \left\{z_{i(t)t+1}\right\} + \left\{x_{0t}\ or\ x_{i(t)+1\ t}\right\}\right\}$
   - $Else\ \left\{first\ candidate\ basis = B(D) - \left\{y_{i(t)t-1}\right\} + \left\{x_{0t}\ or\ x_{i(t)+1\ t}\right\}\right\}.$

2. For any binding constraint from set $A_3$, determine the new candidates as follows:

**Case 1:** ***t=mT***

$$\left\{first\ candidate\ basis = B(D) - \left\{x_{i(t)t}\right\} + \left\{y_{i(t)-1\ t-1}\right\}\right\}$$

**Case 2:** ***1<t<mT***

$$\left\{first\ candidate\ basis = B(D) - \left\{x_{i(t)t}\right\} + \left\{y_{i(t)-1\ t-1}\right\}\right\};$$

- $If\ \left(y_{jt} \notin B(D), for\ j = 1, \dots, n\right)\left\{second\ candidate\ basis = B(D) - \left\{x_{i(t)t}\right\} + \left\{z_{i(t)-1\ t+1}\right\}\right\}.$

**Case 3:** ***t=1***

$$\left\{first\ candidate\ basis = B(D) - \left\{x_{i(t)t}\right\} + \left\{z_{i(t)-1\ t+1}\right\}\right\}.$$

3. For any binding constraint from set $A_1$, determine the new candidates as follows:

**Case 1:** ***t=mT***

$$If\ \left(z_{jt} \notin B(D), for\ j = 1, \dots, n\right)\{first\ candidate\ basis = B(D) - \{x_{0T}\} + \{y_{nT-1}\}\}$$

$$Else\ \left\{first\ candidate\ basis = B(D) - \left\{z_{a(t)\tau(t)+1}\right\} + \left\{y_{a(t)\tau(t)-1}\right\}\right\}.$$

**Case 2:** ***t =mT-1***

$$If\ \left(z_{jt} \notin B(D), for\ j = 1, \dots, n\right)\{first\ candidate\ basis = B(D) - \{x_{0t}\} + \{y_{nt-1}\};$$

- $If\ \left(y_{jt} \notin B(D), for\ j = 1, \dots, n\right) \left\{second\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{n\,t+1}\}\}\right\}$

$Else$ {

- $If\ \left(y_{jt} \notin B(D), for\ j = 1, \dots, n\right)\{first\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{n\,t+1}\};$
    - $If\ (\tau(t) > 1) \left\{second\ candidate\ basis = B(D) - \left\{z_{a(t)\tau(t)+1}\right\} + \left\{y_{a(t)\tau(t)-1}\right\}\}\right\}$
- $Else\ \left\{first\ candidate\ basis = B(D) - \left\{z_{a(t)\tau(t)+1}\right\} + \left\{y_{a(t)\tau(t)-1}\right\}\}\right\}.$

**Case 3:** ***2<t<mT-1***

$If\ \left(z_{jt} \notin B(D), for\ j = 1, \dots, n\right)\{first\ candidate\ basis = B(D) - \{x_{0t}\} + \{y_{nt-1}\};$

- $If\ \left(y_{jt} \notin B(D), for\ j = 1, \dots, n\right) \left\{second\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{n\,t+1}\}\}\right\}$
- $Else$ {
    - $If\ (\tau'(t) < T) \left\{second\ candidate\ basis = B(D) - \left\{y_{b(t)\tau'(t)-1}\right\} + \left\{z_{b(t)\tau'(t)+1}\right\}\}\right\}$

$Else$ {

- $If\ \left(y_{jt} \notin B(D), for\ j = 1, \dots, n\right)\{first\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{n\,t+1}\};$
    - $If\ (\tau(t) > 1) \left\{second\ candidate\ basis = B(D) - \left\{z_{a(t)\tau(t)+1}\right\} + \left\{y_{a(t)\tau(t)-1}\right\}\}\right\}$
- $Else$ {

- $If\ (\tau(t) > 1)\{first\ candidate\ basis = B(D) - \{z_{a(t)\tau(t)+1}\} + \{y_{a(t)\tau(t)-1}\};$
    - $If\ (\tau'(t) < T)\left\{second\ candidate\ basis = B(D) - \{y_{b(t)\tau'(t)-1}\} + \{z_{b(t)\tau'(t)+1}\}\}\right\}$
- $If\ (\tau(t) = 1)\left\{first\ candidate\ basis = B(D) - \{y_{b(t)\tau'(t)-1}\} + \{z_{b(t)\tau'(t)+1}\}\}\right\}.$

**Case 4:** ***t=2***

$$If\ (z_{jt} \notin B(D), for\ j = 1, \dots, n)\{first\ candidate\ basis = B(D) - \{x_{0t}\} + \{y_{nt-1}\};$$

- $If\ (y_{jt} \notin B(D), for\ j = 1, \dots, n)\left\{second\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{n\,t+1}\}\}\right\}$
- $Else\ \{$
    - $If\ (\tau'(t) < T)\left\{second\ candidate\ basis = B(D) - \{y_{b(t)\tau'(t)-1}\} + \{z_{b(t)\tau'(t)+1}\}\}\right\}$

$Else\ \{$

- $If\ (y_{jt} \notin B(D), for\ j = 1, \dots, n)\left\{first\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{n\,t+1}\}\}\right\}$
- $Else\ \left\{first\ candidate\ basis = B(D) - \{y_{b(t)\tau'(t)-1}\} + \{z_{b(t)\tau'(t)+1}\}\}\right\}.$

**Case 5:** ***t=1***

$$If\ (y_{jt} \notin B(D), for\ j = 1, \dots, n)\{first\ candidate\ basis = B(D) - \{x_{0t}\} + \{z_{nt+1}\}\}$$

$$Else\ \left\{first\ candidate\ basis = B(D) - \{y_{b(t)\tau'(t)-1}\} + \{z_{b(t)\tau'(t)+1}\}\right\}.$$

## 4.3. Implementation

In this Section, two algorithms are presented to implement the solution strategy: Semi-Random and Greedy. Because of the some similarities between the two algorithms, just the Semi-Random is discussed in detail.

### 4.3.1. Semi-Random Algorithm "SRA"

In this algorithm after the initialization step, which starts with a specific basis and proposes some other candidates to change the specific basis, the suggested candidates will be visited level by level. In other words, in each level, all the proposed candidates will be checked completely before going to the next level. In fact, we explore the search tree in a breadth-first search (BFS) manner.

In order to detail the algorithm's steps, we need to define some further parameters and variables:

**Parameters**

*$P_{jb}(D_{jb})$ :* the price function of product *j* , which is the inverse of demand intensity function. For instance: $P_{jb}(D_j) = a_{jb} - b_{jb}D_{jb}$.

**Variables**

*B(D) :* set of $x_{jt}$, $y_{jt}$ and $z_{jt}$ variables chosen as basic variables

*B'(D) :* as an indicator of a basis that is being currently tested.

*Ω(B(D)) :* set of linear constraints induced by *B(D)* in terms of $D_{jb}$*s.*

This is defined by the procedure detailed in Section 3.1.

*C(D):* cost minimization objective function, which is formulated as:

$$\sum_{b=1}^{b=m} \sum_{j=1}^{j=n} \sum_{t=bT-T+1}^{t=bT} (c_{jb}x_{jt} + h_{jb}y_{jt} + s_{jb}z_{jt}).$$

*π(D):* non-linear problem's objective function, which is structured as:

$$\sum_{b=1}^{b=m} \sum_{j=1}^{j=n} R_{jb}(D) \sum_{t=bT-T+1}^{t=bT} \beta_{jt} - C(D) .$$

*LCS :* set of linear constraints as follow:

$$\begin{cases} D_{jt} = x_{jt} + y_{jt-1} + z_{jt+1} - y_{jt} - z_{jt}, for\ t = 1, \dots, mT\ and\ j = 1, \dots, n, \\ \sum_{j=0}^{n} x_{jt} = K_t, for\ t = 1, \dots, mT. \end{cases}$$

*V :* set of visited basis

Note that every basis in this set has a *π(D)* value , which can be obtained by solving the *NLP* subject to that basis.

$U_i$ *:* set of non-visited basis in the $i^{th}$ iteration.

*POS :* matrix of potential optimal solutions *($D^*$)* along with corresponding basis and *π(D)* value .

$$POS = \begin{pmatrix} D^* & B(D) & \pi(D) \\ \vdots & \vdots & \vdots \\ D_i^* & B_i(D) & \pi_i(D) \end{pmatrix}$$

**Main body of SRA**

Initialize by *i=0 and B(D)= B'(D)= V= $U_i$= POS =* $\phi$

1. *B'(D)={$x_{jt}$; j=0,…,n and t=1,…,mT}= $B_0$(D).*

If a variable$\notin B(D)$ , it means that it has a zero value.

2. Calculate $x_{jt}$,$y_{jt}$ and $z_{jt}$ values in terms of $D_{jb}$*s* by using the *LCS*. Calculate the *C(D)* value for above found $x_{jt}$, $y_{jt}$ and $z_{jt}$ in terms of $D_{jb}$*s* by using its formulation. Define the *π(D)* in terms of $D_{jb}$*s* by using its structure definition.
3. Apply the procedure defined in Section 4.2.1 to create the set of constraints, *Ω'(B(D))* induced by B'(D).
4. Solve the nonlinear problem of:

   Maximise *π(D),* structured in step 2,

   subject to:

   *Ω'(B(D))* defined in step 3 and

   $\sum_{b=1}^{b=m} \sum_{j=1}^{j=n} \sum_{t=bT-T+1}^{t=bT} D_{jb}\beta_{jt} \leq \sum_{t=1}^{mT} K_t$

This step is in fact optimizing a nonlinear concave objective function over a linear constraint set which guarantees achieving an optimal solution.

5. Add *B'(D)* and it's corresponding *π(D)* to *V.* i.e*., V=V+B'(D).*
6. Check whether or not there is any binding constraint in *Ω'(B(D))* subject to *D**. If NO, *D** is the optimal solution of the problem. Go to step 13. If YES, put *i=i+1* and go to step 7.
7. Apply the procedure defined in Section 3.2 to find the candidate basis for each binding constraint and label them as $B_1(D)$, $B_2(D)$, $B_3(D)$, …

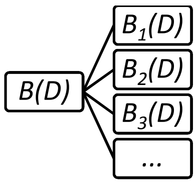



*Figure 4.1. Labelling the found candidate bases*

Note that the parent of all candidates found is the basis which was recently being tested, *B'(D)*.

8. Add all candidates found in step 7 to $U_i$.

$$U_i = \{B_1(D), B_2(D), B_3(D), \dots\}$$

9. Choose one of the basis in set $U_i$ at random; suppose that it is $B_c(D)$. For the chosen basis, do the following steps:
   *9.1.* $B'(D)=B_c(D)$
   *9.2.* Repeat steps 2 and 3 for this new basis, *B'(D)*.
   *9.3.* If the created *Ω'(B(D))* has feasible area, repeat step *4* for this new basis and go to step 9.4. Otherwise
      *9.3.1.* Add parent of *B'(D)* and it's corresponding *D** and *π(D)* to *POS*.
      *9.3.2.* Add *B'(D)* to *V,* give a zero value to the basis in set *V,* delete *B'(D)* from $U_i$ , *i=i+1* and go to step 10 .
   *9.4.* Add *B'(D)* and it's corresponding *π(D)* to *V,* delete *B'(D)* from $U_i$.
   *9.5.* Check whether or not there is any binding constraint in *Ω'(B(D))* subject to *D**. If NO, Add *B'(D)* and it's corresponding *D** and *π(D)* to *POS*. *i=i+1*, go to step 10. If YES, *i=i+1.*

9.6. Apply the procedure defined in Section 4.2.2 to find the candidate basis for each binding constraint and label them as $B_{c1}(D)$, $B_{c2}(D)$, $B_{c3}(D)$, …

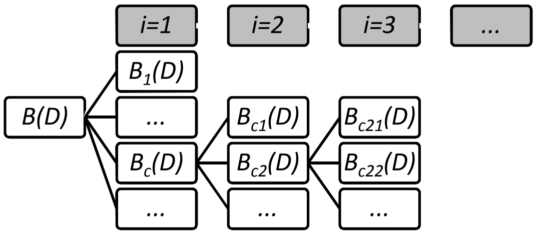



*Figure 4.2. Labelling the found candidate bases for each binding constraint*

Note that the number of index in *B(D)* is equal to *i*. For example in the first iteration we have $B_1(D)$, $B_2(D)$, $B_3(D)$, … , in the second iteration we have $B_{11}(D)$, $B_{12}(D)$, $B_{13}(D)$, … and in the fifth iteration we have $B_{12312}(D)$, $B_{21342}(D)$, $B_{42315}(D)$, …

9.7. Add those candidates found in step 9.6 which are not equal to none of the elements of set *V* and $U_1, U_2, \ldots, U_i$ to $U_j$ . Update *POS* by those candidates found in step 9.6 which are already in set *V or* $U_1, U_2, \ldots, U_i$ to $U_j$.

10. *i=i-1* and restart from step *9*; continue till set $U_j$ comes to $\phi$ .

11. *i=i+1* , check if $U_j$ exists and is not $\phi$ go to step 9, if it doesn't exist or is $\phi$ go to 12.

12. Find the biggest value of *π(D)* in *POS*, and show the corresponding $D^*$ and basis for that.

13. Calculate the optimal price for each product by using the price function, $P_j(D_j)$. Show the values of x, y, z variables corresponding to the optimal basis.

The following flowchart summarises the main steps of the algorithm.

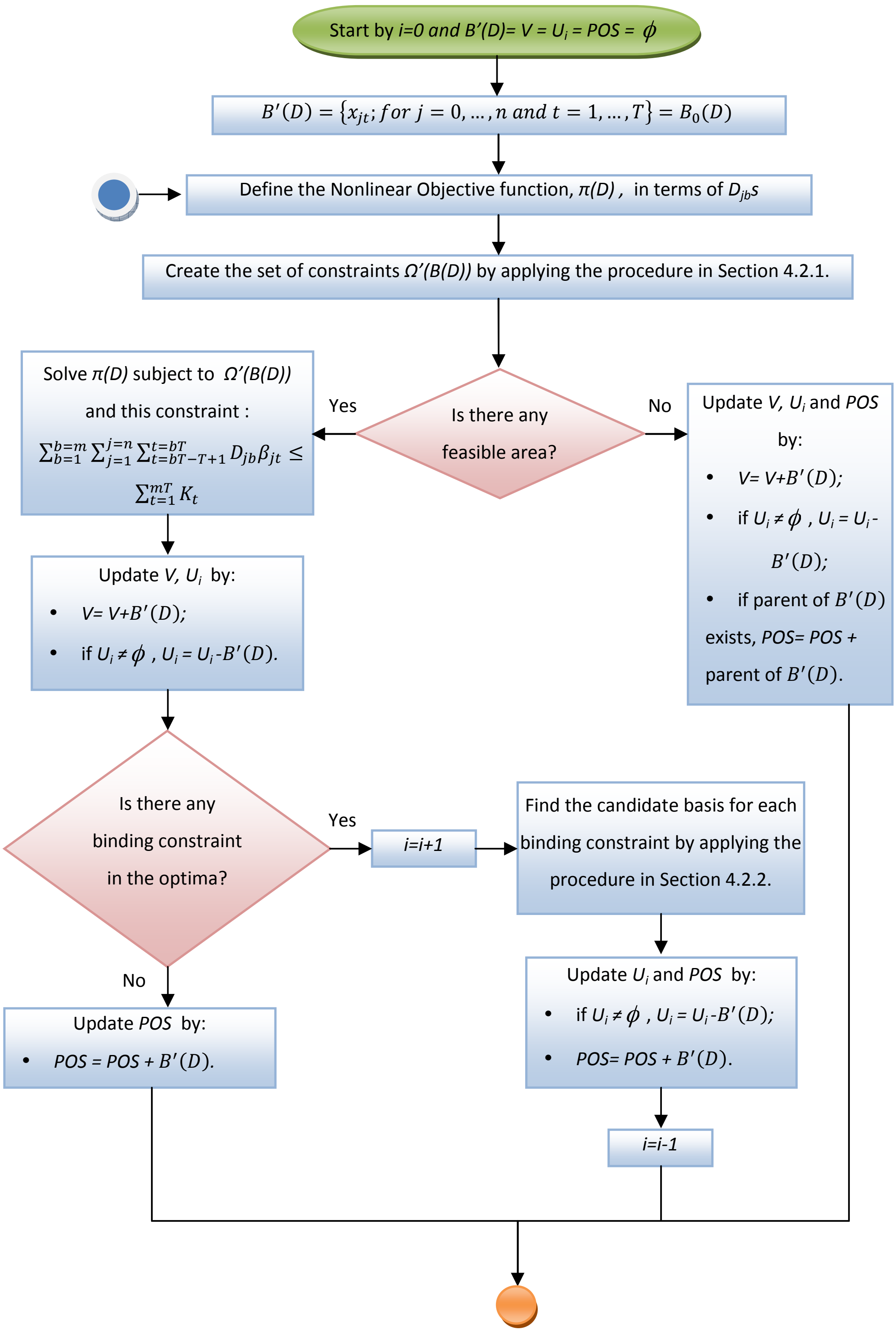



*Figure 4.3. Flowchart of the SRA*

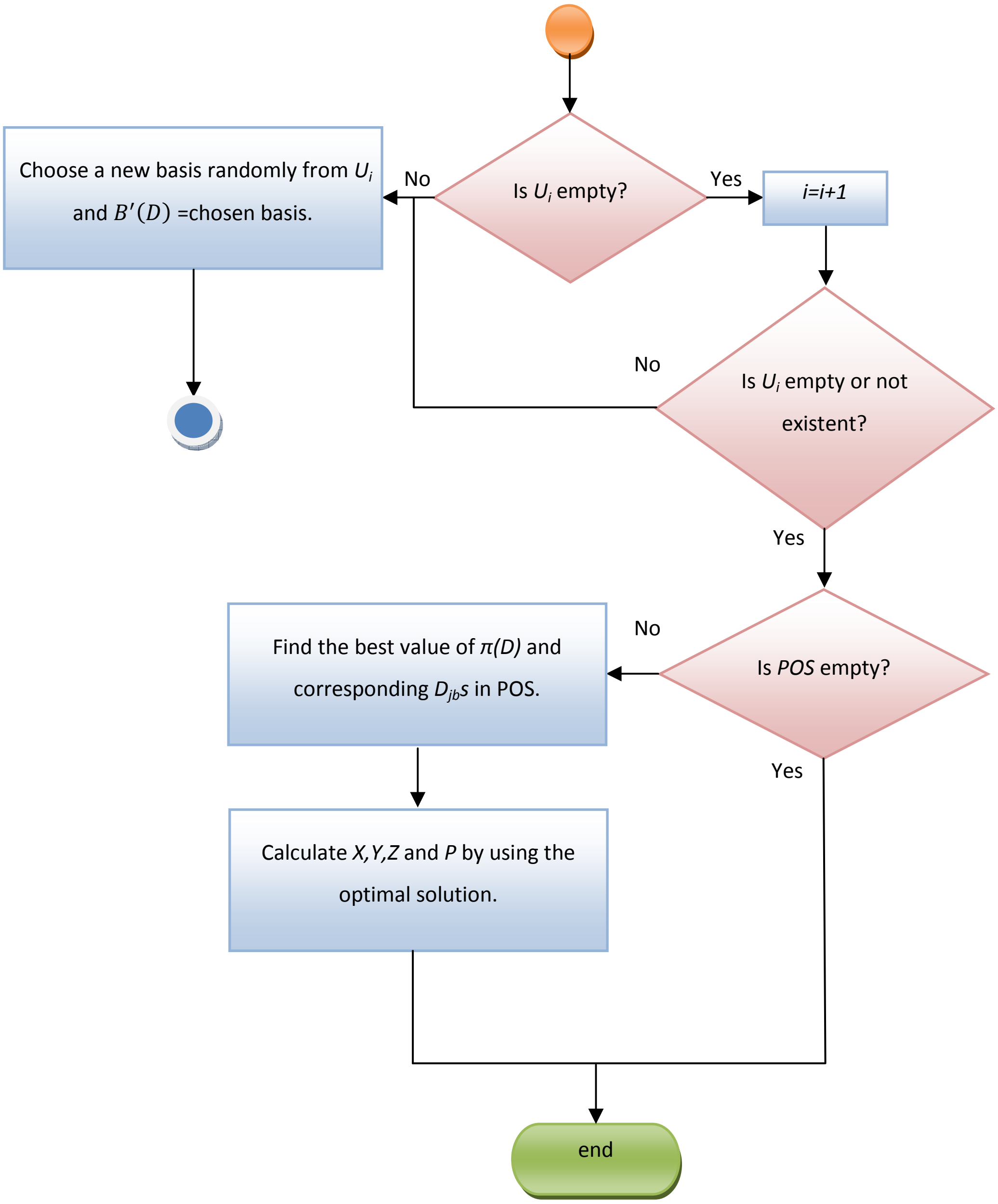



*Figure 4.3b. Flowchart of the SRA (continue)*

## 4.3.2. Greedy algorithm

In this algorithm, the only important dissimilarity with the SRA is that we don't need to define set $U_i$ in each level as the set of non-visited basis.

Regardless of the level of the visited basis, each basis which has a greater value for the objective function will be chosen earlier to find its next level.

### 4.3.3. Numerical Example

In order to display our algorithm more clearly, consider a case with *n=2* products and *T =12* periods. The parameters for the two products are as shown in Table 4.1.

*Table 4.1. Parameters of the dynamic pricing example with n=2 products and T =12 periods*

| | **b** | **1** | | | **2** | | | **3** | | | **4** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **t** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| $\beta_{jt}$ | **j=1** | 0.6 | 0.4 | 1 | 1.1 | 1 | 0.9 | 0.4 | 0.7 | 0.4 | 0.5 | 0.8 | 1.2 |
| | **j=2** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $c_{jb}$ | **j=1** | 10 | 10 | 10 | 9 | 9 | 9 | 11 | 11 | 11 | 12 | 12 | 12 |
| | **j=2** | 8 | 8 | 8 | 7 | 7 | 7 | 8 | 8 | 8 | 9 | 9 | 9 |
| $h_{jb}$ | **j=1** | 6 | 6 | 6 | 5 | 5 | 5 | 4 | 4 | 4 | 7 | 7 | 7 |
| | **j=2** | 3 | 3 | 3 | 4 | 4 | 4 | 2 | 2 | 2 | 5 | 5 | 5 |
| $s_{jb}$ | **j=1** | 8 | 8 | 8 | 9 | 9 | 9 | 7 | 7 | 7 | 8 | 8 | 8 |
| | **j=2** | 6 | 6 | 6 | 7 | 7 | 7 | 4 | 4 | 4 | 6 | 6 | 6 |
| $D_{jt}$ | **j=1** | $0.6\ D_{11}$ | $0.4\ D_{11}$ | $D_{11}$ | $1.1\ D_{12}$ | $D_{12}$ | $0.9\ D_{12}$ | $0.4\ D_{13}$ | $0.7\ D_{13}$ | $0.4\ D_{13}$ | $0.5\ D_{14}$ | $0.8\ D_{14}$ | $1.2\ D_{14}$ |
| | **j=2** | $D_{21}$ | $D_{21}$ | $D_{21}$ | $D_{22}$ | $D_{22}$ | $D_{22}$ | $D_{23}$ | $D_{23}$ | $D_{23}$ | $D_{24}$ | $D_{24}$ | $D_{24}$ |
| $p_{jb}$ | **j=1** | $30 - 0.2\ D_{11}$ | | | $30 - 0.2\ D_{12}$ | | | $30 - 0.2\ D_{13}$ | | | $30 - 0.2\ D_{14}$ | | |
| | **j=2** | $30 - 0.2\ D_{21}$ | | | $30 - 0.2\ D_{22}$ | | | $30 - 0.2\ D_{23}$ | | | $30 - 0.2\ D_{24}$ | | |
| $K_t$ | | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

Note that the relationship between price and demand intensity for both products is identical, and that their cross price elasticity is zero. For ease of manual computation, we assume a fixed production capacity, $K_t$ =I00, for all periods in the planning horizon. In order to solve this example we choose the SRA and follow it step by step.

Initialize by *i=0 and B(D)= B'(D)= V= U$_i$= POS =* $\phi$.

**1.**

$$B(D) = \{x_{01}, x_{02}, x_{03}, x_{04}, x_{05}, x_{06}, x_{07}, x_{08}, x_{09}, x_{010}, x_{011}, x_{012},\\ x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{110}, x_{111}, x_{112},\\ x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}, x_{27}, x_{28}, x_{29}, x_{210}, x_{211}, x_{212}\}$$

and *B'(D)=B(D).*

**2.**

$$x_{11}= 0.6D_{11}, x_{12}= 0.4D_{11}, x_{13}= D_{11}, x_{14}= 1.1D_{12}, x_{15}= D_{12}, x_{16}= 0.9D_{12},$$
$$x_{17}= 0.4D_{13}, x_{18}= 0.7D_{13}, x_{19}= 0.4D_{13}, x_{110}= 0.5D_{14}, x_{111}= 0.8D_{14}, x_{112}= 1.2D_{14},$$
$$x_{21}=x_{22}=x_{23}= D_{21}, x_{24}=x_{25}=x_{26}= D_{22}, x_{27}=x_{28}=x_{29}= D_{23}, x_{210}=x_{211}=x_{212}= D_{24},$$
$$x_{01}= 100-0.6D_{11}-D_{21}, x_{02}= 100-0.4D_{11}-D_{21}, x_{03}= 100-D_{11}-D_{21},$$
$$x_{04}= 100-1.1D_{12}-D_{22}, x_{05}= 100-D_{12}-D_{22}, x_{06}= 100-0.9D_{12}-D_{22}, x_{07}= 100-0.4D_{13}-D_{23},$$
$$x_{08}= 100-0.7D_{13}-D_{23}, x_{09}= 100-0.4D_{13}-D_{23}, x_{010}= 100-0.5D_{14}-D_{24},$$
$$x_{011}= 100-0.8D_{14}-D_{24}, x_{012}= 100-1.2D_{14}-D_{24}.$$

Note that all other *y* and *z* variables are zero.

$$C(D) = [10(x_{11}+x_{12}+x_{13})+9(x_{14}+x_{15}+x_{16})+11(x_{17}+x_{18}+x_{19})+12(x_{110}+x_{111}+x_{112})+\\ 8(x_{21}+x_{22}+x_{23})+7(x_{24}+x_{25}+x_{26})+8(x_{27}+x_{28}+x_{29})+9(x_{210}+x_{211}+x_{212})+\\ 6(y_{11}+y_{12}+y_{13})+5(y_{14}+y_{15}+y_{16})+4(y_{17}+y_{18}+y_{19})+7(y_{110}+y_{111})+\\ 3(y_{21}+y_{22}+y_{23})+4(y_{24}+y_{25}+y_{26})+2(y_{27}+y_{28}+y_{29})+5(y_{210}+y_{211})+\\ 8(z_{12}+z_{13})+9(z_{14}+z_{15}+z_{16})+7(z_{17}+z_{18}+z_{19})+8(z_{110}+z_{111}+z_{112})+6(z_{22}+z_{23})+\\ 7(z_{24}+z_{25}+z_{26})+4(z_{27}+z_{28}+z_{29})+6(z_{210}+z_{211}+z_{212})] =\\ 20D_{11}+24D_{21}+27D_{12}+21D_{22}+16.5D_{13}+24D_{23}+30D_{14}+27D_{24}.$$

The NLP structure is:

$$\sum_{b=1}^{b=4}\sum_{j=1}^{2} P_{jb}(D_{jb}).D_{jb}\sum_{t=1}^{12}\beta_{jt} - C(D) = 2D_{11}(30-0.2D_{11})+3D_{21}(30-0.2D_{21})+\\ 3D_{12}(30-0.2D_{12})+3D_{22}(30-0.2D_{22})+1.5D_{13}(30-0.2D_{13})+3D_{23}(30-0.2D_{23})+\\ 2.5D_{14}(30-0.2D_{14})+3D_{24}(30-0.2D_{24})-\\ (20D_{11}+24D_{21}+27D_{12}+21D_{22}+16.5D_{13}+24D_{23}+30D_{14}+27D_{24}) =\\ -0.4D_{11}^2-0.6D_{21}^2-0.6D_{12}^2-0.6D_{22}^2-0.3D_{13}^2-0.6D_{23}^2-0.5D_{14}^2-0.6D_{24}^2+40D_{11}+66D_{21}+\\ 63D_{12}+69D_{22}+28.5D_{13}+66D_{23}+45D_{14}+63D_{24}.$$

**3.**

$$A_1 = \begin{Bmatrix} t=1; & x_{01} \in B(D) & \rightarrow & 0.6D_{11} + D_{21} \le 100 \\ t=2; & x_{02} \in B(D) & \rightarrow & 0.4D_{11} + D_{21} \le 100 \\ t=3; & x_{03} \in B(D) & \rightarrow & D_{11} + D_{21} \le 100 \\ t=4; & x_{04} \in B(D) & \rightarrow & 1.1D_{12} + D_{22} \le 100 \\ t=5; & x_{05} \in B(D) & \rightarrow & D_{12} + D_{22} \le 100 \\ t=6; & x_{06} \in B(D) & \rightarrow & 0.9D_{12} + D_{22} \le 100 \\ t=7; & x_{07} \in B(D) & \rightarrow & 0.4D_{13} + D_{23} \le 100 \\ t=8; & x_{08} \in B(D) & \rightarrow & 0.7D_{13} + D_{23} \le 100 \\ t=9; & x_{09} \in B(D) & \rightarrow & 0.4D_{13} + D_{23} \le 100 \\ t=10; & x_{010} \in B(D) & \rightarrow & 0.5D_{14} + D_{24} \le 100 \\ t=11; & x_{011} \in B(D) & \rightarrow & 0.8D_{14} + D_{24} \le 100 \\ t=12; & x_{012} \in B(D) & \rightarrow & 1.2D_{14} + D_{24} \le 100 \end{Bmatrix}$$

$A_2, A_3 = \phi$

$\Omega'(B(D)) = A_1$.

$$\begin{aligned} &Max_{D\ge 0}\{-0.4D_{11}^2 - 0.6D_{21}^2 - 0.6D_{12}^2 - 0.6D_{22}^2 - 0.3D_{13}^2 - 0.6D_{23}^2 - \\ &\quad 0.5D_{14}^2 - 0.6D_{24}^2 + 40D_{11} + 66D_{21} + 63D_{12} + 69D_{22} \\ &\quad + 28.5D_{13} + 66D_{23} + 45D_{14} + 63D_{24}\} \\ &subject\ to: \\ &t=1; \quad 0.6D_{11} + D_{21} \le 100 \\ &t=2; \quad 0.4D_{11} + D_{21} \le 100 \\ &t=3; \quad D_{11} + D_{21} \le 100 \\ &t=4; \quad 1.1D_{12} + D_{22} \le 100 \\ &t=5; \quad D_{12} + D_{22} \le 100 \\ &t=6; \quad 0.9D_{12} + D_{22} \le 100 \\ &t=7; \quad 0.4D_{13} + D_{23} \le 100 \\ &t=8; \quad 0.7D_{13} + D_{23} \le 100 \\ &t=9; \quad 0.4D_{13} + D_{23} \le 100 \\ &t=10; \quad 0.5D_{14} + D_{24} \le 100 \\ &t=11; \quad 0.8D_{14} + D_{24} \le 100 \\ &t=12; \quad 1.2D_{14} + D_{24} \le 100 \quad and \end{aligned}$$

**4.** $2D_{11} + 3D_{12} + 1.5D_{13} + 2.5D_{14} + 3D_{21} + 3D_{22} + 3D_{23} + 3D_{24} \le 1200$.

$$\begin{aligned} &Optimal\ \ Solution: (D^*_{11}, D^*_{21}, D^*_{12}, D^*_{22}, D^*_{13}, D^*_{23}, D^*_{14}, D^*_{24}) = \\ &(47, 53, 44.9, 50.6, 47.5, 55, 41.5, 50.1) \quad and \quad \pi(D^*) = 11532.2 \end{aligned}$$

**5.** $V = \{(B(D), 11532.2)\}$.

Note that the corresponding objective value is given to each element in set *V.*

**6.** $t=3;\ D_{11} + D_{21} \le 100$ , $t=4;\ 1.1D_{12} + D_{22} \le 100$ and $t=12;\ 1.2D_{14} + D_{24} \le 100$ from set $A_1$ subject to $D^*$ are binding. *i=1.*

**7.** Since the binding constraints belong to set $A_1$, the candidate bases for them are:

$B_1(D)=B(D)-\{x_{03}\}+\{y_{22}\}=\{...,y_{22}\}$ $B_2(D)=B(D)-\{x_{03}\}+\{z_{24}\}=\{...,z_{24}\}$
$B_3(D)=B(D)-\{x_{04}\}+\{y_{23}\}=\{...,y_{23}\}$ $B_4(D)=B(D)-\{x_{04}\}+\{z_{25}\}=\{...,z_{25}\}$
$B_5(D)=B(D)-\{x_{012}\}+\{y_{211}\}=\{...,y_{211}\}$.

**8.** $U_1=\{B_1(D),B_2(D),B_3(D),B_4(D),B_5(D)\}$.

**9.** Suppose that the algorithm chooses *B₁(D)* at random.

**9.1.** $B'(D)=B_1(D)=\{...,y_{22}\}$.

**9.2.**

$x_{11}=0.6D_{11}, x_{12}=0.4D_{11}, x_{13}=D_{11}, x_{14}=1.1D_{12}, x_{15}=D_{12}, x_{16}=0.9D_{12},$
$x_{17}=0.4D_{13}, x_{18}=0.7D_{13}, x_{19}=0.4D_{13}, x_{110}=0.5D_{14}, x_{111}=0.8D_{14},$
$x_{112}=1.2D_{14}, x_{21}=D_{21}, x_{22}=2D_{21}+D_{11}-100, x_{23}=100-D_{11},$
$x_{24}=x_{25}=x_{26}=D_{22}, x_{27}=x_{28}=x_{29}=D_{23}, x_{210}=x_{211}=x_{212}=D_{24},$
$x_{01}=100-0.6D_{11}-D_{21}, x_{02}=200-1.4D_{11}-2D_{21}, y_{22}=D_{11}+D_{21}-100,$
$x_{04}=100-1.1D_{12}-D_{22}, x_{05}=100-D_{12}-D_{22}, x_{06}=100-0.9D_{12}-D_{22},$
$x_{07}=100-0.4D_{13}-D_{23}, x_{08}=100-0.7D_{13}-D_{23}, x_{09}=100-0.4D_{13}-D_{23},$
$x_{010}=100-0.5D_{14}-D_{24}, x_{011}=100-0.8D_{14}-D_{24}, x_{012}=100-1.2D_{14}-D_{24}.$

Note that $x_{03}$ is a non-basic, and $y_{22}$ is a basic variable.

$C(D)=23D_{11}+27D_{21}+27D_{12}+21D_{22}+16.5D_{13}+24D_{23}+30D_{14}+27D_{24}-300.$

The NLP structure is:

$$-0.4D_{11}^2-0.6D_{21}^2-0.6D_{12}^2-0.6D_{22}^2-0.3D_{13}^2-0.6D_{23}^2-0.5D_{14}^2-0.6D_{24}^2+$$
$$37D_{11}+63D_{21}+63D_{12}+69D_{22}+28.5D_{13}+66D_{23}+45D_{14}+63D_{24}+300.$$

$$A_1=\left\{\begin{array}{lllll} t=1; & x_{01}\in B(D) & \rightarrow & 0.6D_{11}+D_{21}\le 100 \\ t=2; & x_{02}\in B(D) & \rightarrow & 1.4D_{11}+2D_{21}\le 200 \\ t=4; & x_{04}\in B(D) & \rightarrow & 1.1D_{12}+D_{22}\le 100 \\ t=5; & x_{05}\in B(D) & \rightarrow & D_{12}+D_{22}\le 100 \\ t=6; & x_{06}\in B(D) & \rightarrow & 0.9D_{12}+D_{22}\le 100 \\ t=7; & x_{07}\in B(D) & \rightarrow & 0.4D_{13}+D_{23}\le 100 \\ t=8; & x_{08}\in B(D) & \rightarrow & 0.7D_{13}+D_{23}\le 100 \\ t=9; & x_{09}\in B(D) & \rightarrow & 0.4D_{13}+D_{23}\le 100 \\ t=10; & x_{010}\in B(D) & \rightarrow & 0.5D_{14}+D_{24}\le 100 \\ t=11; & x_{011}\in B(D) & \rightarrow & 0.8D_{14}+D_{24}\le 100 \\ t=12; & x_{012}\in B(D) & \rightarrow & 1.2D_{14}+D_{24}\le 100 \end{array}\right\}$$

$$A_2=\{t=3;\quad x_{03}\notin B(D)\quad \rightarrow\quad D_{11}+D_{21}>100\}$$
$$A_3=\{t=3;\quad x_{03}\notin B(D)\quad \rightarrow\quad D_{11}\le 100\}$$
$$\Omega'(B(D))=A_1\cup A_2\cup A_3.$$

**9.3.** *Ω'(B(D))* has feasible area, solve the *NLP*:

**9.4.**

$$Max_{D\geq 0}\{-0.4D_{11}^2-0.6D_{21}^2-0.6D_{12}^2-0.6D_{22}^2-0.3D_{13}^2-0.6D_{23}^2-$$
$$0.5D_{14}^2-0.6D_{24}^2+37D_{11}+63D_{21}+63D_{12}+69D_{22}+$$
$$28.5D_{13}+66D_{23}+45D_{14}+63D_{24}+300\}$$

$subject\ \ to:$

$t=1;\qquad 0.6D_{11}+D_{21}\leq 100$

$t=2;\qquad 1.4D_{11}+2D_{21}\leq 200$

$t=3;\qquad D_{11}+D_{21}>100$

$t=3;\qquad D_{11}\leq 100$

$t=4;\qquad 1.1D_{12}+D_{22}\leq 100$

$t=5;\qquad D_{12}+D_{22}\leq 100$

$t=6;\qquad 0.9D_{12}+D_{22}\leq 100$

$t=7;\qquad 0.4D_{13}+D_{23}\leq 100$

$t=8;\qquad 0.7D_{13}+D_{23}\leq 100$

$t=9;\qquad 0.4D_{13}+D_{23}\leq 100$

$t=10;\qquad 0.5D_{14}+D_{24}\leq 100$

$t=11;\qquad 0.8D_{14}+D_{24}\leq 100$

$t=12;\qquad 1.2D_{14}+D_{24}\leq 100\qquad and$

$2D_{11}+3D_{12}+1.5D_{13}+2.5D_{14}+3D_{21}+3D_{22}+3D_{23}+3D_{24}\leq 1200\,.$

$Optimal\ \ Solution:(D^*_{11},D^*_{21},D^*_{12},D^*_{22},D^*_{13},D^*_{23},D^*_{14},D^*_{24})=$
$(47\ ,\ 53\ ,\ 44.9\ ,\ 50.6\ ,\ 47.5\ ,\ 55\ ,\ 41.5\ ,\ 50.1)\qquad and\quad \pi(D^*)=11532.2$

**9.5.** $V=\{(B(D),11532\ .2),(B_1(D),11532\ .2)\}$ and

$U_1=\{B_2(D),B_3(D),B_4(D),B_5(D)\}.$

**9.6.** $t=3;\quad D_{11}+D_{21}>100$ from set $A_2$ and $t=4;\quad 1.1D_{12}+D_{22}\leq 100$ and $t=12;\quad 1.2D_{14}+D_{24}\leq 100$ from set $A_1$ subject to $D^*$ are binding. *i=2.*

**9.7.** For the binding constraint which belongs to set $A_2$ , the candidate basis is defined as follow:

$B_{11}(D)=B'(D)-\{y_{22}\}+\{x_{03}\}=\{\forall x_{jt}\ \ ,for\ j=0,1,2\ and\ t=1,...,12\}.$

For the binding constraints which belong to set $A_1$, the candidate bases are:

$B_{12}(D)=B(D)-\{x_{04}\}+\{y_{23}\}=\{...y_{22},y_{23}\}$
$B_{13}(D)=B(D)-\{x_{04}\}+\{z_{25}\}=\{...y_{22},z_{25}\}$
$B_{14}(D)=B(D)-\{x_{012}\}+\{y_{211}\}=\{...y_{22},y_{211}\}.$

**9.8.** Candidate $B_{11}(D)$ is like *B (D)* in set *V*. So the matrix POS and set $U_2$ are constructed as :

$$POS = \left( \begin{array}{c|c|c} \textit{Decisions' Value} & \textit{Basis} & \textit{Objective Value} \\ \hline (47\,,\,53\,,\,44.9\,,\,50.6\,,\,47.5\,,\,55\,,\,41.5\,,\,50.1) & B(D) & 11532.2 \end{array} \right)$$

$$U_2 = \{B_{12}(D), B_{13}(D), B_{14}(D)\}$$

**10.** *i=1* and restart from step 9; continue till set $U_1$ comes to $\phi$ .

Due to the limitations in bringing the long computational details in the thesis, we summarise the result of the remaining steps in Table 4.2. Continuing in this way finds the following candidate bases which have been visited and shown.

*Table 4.2. List of visited basis stemmed from the continued algorithm.*

| **The Visited Basis** | **Elements of the Basis** |
|---|---|
| *$B_3(D)$* | $\{x_{01},x_{02},x_{03},y_{23},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},x_{012},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$<br>$x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| *$B_5(D)$* | $\{x_{01},x_{02},x_{03},x_{04},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},y_{211},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$<br>$x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| *$B_2(D)$* | $\{x_{01},x_{02},z_{24},x_{04},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},x_{012},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$<br>$x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| *$B_4(D)$* | $\{x_{01},x_{02},x_{03},z_{25},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},x_{012},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$<br>$x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| *$B_{12}(D)$* | $\{x_{01},x_{02},y_{22},y_{23},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},x_{012},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$<br>$x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| *$B_{22}(D)$* | $\{x_{01},x_{02},z_{24},z_{25},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},x_{012},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$<br>$x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| *$B_{14}(D)$* | $\{x_{01},x_{02},y_{22},x_{04},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},y_{211},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$<br>$x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| *$B_{13}(D)$* | $\{x_{01},x_{02},y_{22},z_{25},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},x_{012},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$<br>$x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| *$B_{54}(D)$* | $\{x_{01},x_{02},x_{03},z_{25},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},y_{211},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$<br>$x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |

| | |
|---|---|
| $B_{33}(D)$ | $\{x_{01},x_{02},x_{03},y_{23},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},y_{211},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$ $x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| $B_{44}(D)$ | $\{x_{01},x_{02},x_{03},z_{25},z_{26},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},x_{012},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$ $x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| $B_{52}(D)$ | $\{x_{01},x_{02},z_{24},x_{04},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},y_{211},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$ $x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| $B_{143}(D)$ | $\{x_{01},x_{02},y_{22},z_{25},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},y_{211},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$ $x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| $B_{442}(D)$ | $\{x_{01},x_{02},z_{24},z_{25},z_{26},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},x_{012},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$ $x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| $B_{223}(D)$ | $\{x_{01},x_{02},z_{24},z_{25},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},y_{211},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$ $x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| $B_{121}(D)$ | $\{x_{01},x_{02},y_{22},y_{23},x_{05},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},y_{211},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$ $x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| $B_{543}(D)$ | $\{x_{01},x_{02},x_{03},z_{25},z_{26},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},y_{211},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$ $x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| $B_{132}(D)$ | $\{x_{01},x_{02},y_{22},z_{25},z_{26},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},x_{012},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$ $x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| $B_{4423}(D)$ | $\{x_{01},x_{02},z_{24},z_{25},z_{26},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},y_{211},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$ $x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |
| $B_{1432}(D)$ | $\{x_{01},x_{02},y_{22},z_{25},z_{26},x_{06},x_{07},x_{08},x_{09},x_{010},x_{011},y_{211},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},$ $x_{17},x_{18},x_{19},x_{110},x_{111},x_{112},x_{21},x_{22},x_{23},x_{24},x_{25},x_{26},x_{27},x_{28},x_{29},x_{210},x_{211},x_{212}\}$ |

Here we detail some of the final steps of the algorithm:

**1.** This step is in fact the application of the step 9 and its sub steps. Suppose that the algorithm chooses *$B_{1432}(D)$* at random.

**1.1.** $B'(D) = B_{1432}(D) = \{..., y_{22}, z_{25}, z_{26}, y_{211}\}$.

**1.2.**

$x_{11}= 0.6D_{11}, x_{12}= 0.4D_{11}, x_{13}= D_{11}, x_{14}= 1.1D_{12}, x_{15}= D_{12}, x_{16}= 0.9D_{12},$
$x_{17}= 0.4D_{13}, x_{18}= 0.7D_{13}, x_{19}= 0.4D_{13}, x_{110}= 0.5D_{14}, x_{111}= 0.8D_{14}, x_{112}= 1.2D_{14},$
$x_{21}= D_{21}, x_{22}= D_{11} + 2D_{21} - 100, x_{23} = 100 - D_{11}, x_{24}= 100 - 1.1D_{12},$
$x_{25}= 100 - D_{12}, x_{26} = 2.1D_{12} + 3D_{22} - 200, x_{27}=x_{28}=x_{29}= D_{23}, x_{210}= D_{24},$
$x_{211}= 1.2D_{14} + 2D_{24} - 100, x_{212}= 100 - 1.2D_{14}, x_{01}= 100 - 0.6D_{11} - D_{21},$
$x_{02}= 100 - 0.4D_{11} - D_{21}, y_{22}= 100 - D_{11} - D_{21}, z_{25}= 1.1D_{12} + D_{22} - 100,$
$z_{26}= 2.1D_{12} + 2D_{22} - 200, x_{06}= 300 - 3D_{12} - 3D_{22}, x_{07}= 100 - 0.4D_{13} - D_{23},$
$x_{08}= 100 - 0.7D_{13} - D_{23}, x_{09}= 100 - 0.4D_{13} - D_{23}, x_{010}= 100 - 0.5D_{14} - D_{24},$
$x_{011}= 200 - 2D_{14} - 2D_{24}, y_{211}= 1.2D_{14} + D_{24} - 100.$

Note that $x_{03}$ , $x_{04}$ , $x_{05}$ and $x_{012}$ are non-basic, and $y_{22}$ , $z_{25}$ , $z_{26}$ and $y_{211}$ are basic variables.

$$C(D) = 23D_{11} + 27D_{21} + 50.1D_{12} + 42D_{22} + 16.5D_{13} + 24D_{23} + 36D_{14} + 32D_{24} - 2900.$$

The NLP structure is:

$$-0.4D_{11}^2 - 0.6D_{21}^2 - 0.6D_{12}^2 - 0.6D_{22}^2 - 0.3D_{13}^2 - 0.6D_{23}^2 - 0.5D_{14}^2 - 0.6D_{24}^2 +$$
$$37D_{11} + 63D_{21} + 39.9D_{12} + 48D_{22} + 28.5D_{13} + 66D_{23} + 39D_{14} + 58D_{24} + 2900.$$

$$A_1 = \begin{Bmatrix} t = 1; & x_{01} \in B(D) & \to & 0.6D_{11} + D_{21} \le 100 \\ t = 2; & x_{02} \in B(D) & \to & 1.4D_{11} + 2D_{21} \le 200 \\ t = 6; & x_{06} \in B(D) & \to & 3D_{12} + 3D_{22} \le 300 \\ t = 7; & x_{07} \in B(D) & \to & 0.4D_{13} + D_{23} \le 100 \\ t = 8; & x_{08} \in B(D) & \to & 0.7D_{13} + D_{23} \le 100 \\ t = 9; & x_{09} \in B(D) & \to & 0.4D_{13} + D_{23} \le 100 \\ t = 10; & x_{010} \in B(D) & \to & 0.5D_{14} + D_{24} \le 100 \\ t = 11; & x_{011} \in B(D) & \to & 2D_{14} + 2D_{24} \le 200 \end{Bmatrix}$$

$$A_2 = \begin{Bmatrix} t = 3; & x_{03} \notin B(D) & \to & D_{11} + D_{21} > 100 \\ t = 4; & x_{04} \notin B(D) & \to & 1.1D_{12} + D_{22} > 100 \\ t = 5; & x_{05} \notin B(D) & \to & 2.1D_{12} + 2D_{22} > 200 \\ t = 12; & x_{012} \notin B(D) & \to & 1.2D_{14} + D_{24} > 100 \end{Bmatrix}$$

$$A_3 = \begin{Bmatrix} t = 3; & x_{03} \notin B(D) & \to & D_{11} \le 100 \\ t = 4; & x_{04} \notin B(D) & \to & 1.1D_{12} \le 100 \\ t = 5; & x_{05} \notin B(D) & \to & D_{12} \le 100 \\ t = 12; & x_{012} \notin B(D) & \to & 1.2D_{14} \le 100 \end{Bmatrix}$$

$$\Omega'(B(D)) = A_1 \cup A_2 \cup A_3.$$

**1.3.** *Ω'(B(D))* has no feasible area, solve the *NLP*:

**1.3.1.** Parent of *B'(D)* is *$B_{143}(D)$:*

$$POS = \begin{pmatrix} \textit{Decisions Value} & \textit{Basis} & \textit{Objective Value} \\ \hline (47, 53, 44.9, 50.6, 47.5, 55, 41.5, 50.1) & B(D) & 11532.2 \\ (46.2, 52.5, 46.08, 51.6, 47.5, 55, 41.5, 50.1) & B_{12}(D) & 11534.08 \\ (47, 53, 46.05, 51.6, 47.5, 55, 41.5, 50.1) & B_4(D) & 11533.7 \\ (47, 53, 44.9, 50.5, 47.5, 55, 41.5, 50.1) & B_{14}(D) & 11532.2 \\ (45.8, 52.2, 45.8, 51.4, 47.5, 55, 41.5, 50.1 & B_{33}(D) & 11533.9 \\ (47, 53, 44.9, 50.6, 47.5, 55, 41.5, 50.1) & B_1(D) & 11532.2 \\ (47, 53, 44.9, 50.5, 47.5, 55, 41.5, 50.1) & B_{52}(D) & 11532.1 \\ (47, 53, 46.05, 51.6, 47.5, 55, 41.5, 50.1) & B_{13}(D) & 11533.7 \\ (47, 53, 44.6, 50.9, 47.5, 55, 41.5, 50.1) & B_{22}(D) & 11296.3 \\ (47, 53, 46.05, 51.6, 47.5, 55, 41.5, 50.1) & B_{54}(D) & 11533.7 \\ (45.8, 52.2, 45.8, 51.4, 47.5, 55, 41.5, 50.1) & B_3(D) & 11533.9 \\ (47, 53, 44.9, 50.6, 47.5, 55, 41.5, 50.1) & B_2(D) & 11532.2 \\ (47, 53, 44.9, 50.6, 47.5, 55, 41.5, 50.1) & B_5(D) & 11532.2 \\ (46.2, 52.5, 46.08, 51.6, 47.5, 55, 42, 49.5) & B_{121}(D) & 11534.08 \\ (47, 53, 46.05, 51.6, 47.5, 55, 41.5, 50.1) & B_{143}(D) & 11533.7 \\ (47, 53, 44.6, 50.9, 47.5, 55, 41.5, 50.1) & B_{223}(D) & 11296.3 \\ (47.5, 52.4, 45.8, 51.8, 44.5, 52.04, 42.04, 49.5) & B_{132}(D) & 10948.08 \\ (47, 53, 45.7, 51.9, 47.5, 55, 41.5, 50.1) & B_{543}(D) & 11501.5 \\ (47, 53, 45.7, 51.4, 47.5, 55, 41.5, 50.1) & B_{44}(D) & 11501.5 \\ 0 & B_{1432}(D) & 0 \\ 0 & B_{4423}(D) & 0 \\ (47.5, 52.4, 45.8, 51.8, 44.5, 52.04, 42.04, 49.5) & B_{442}(D) & 10948.08 \end{pmatrix}$$

**1.4.** Add *B′(D)* to *V,* delete *B′(D)* from $U_i$ , *i=i+1* and go to step *10* .

$$V = \begin{Bmatrix} (B(D),11532.2),(B_1(D),11532.2),(B_3(D),11533.9),(B_5(D),11532.2), \\ (B_2(D),11532.2)(B_4(D),11533.7),(B_{12}(D),11534.08),(B_{22}(D),11296.3), \\ (B_{14}(D),11532.2),(B_{13}(D),11533.7)(B_{54}(D),11533.7),(B_{33}(D),11533.9), \\ (B_{44}(D),11501.5),(B_{52}(D),11532.1),(B_{143}(D),11533.7)(B_{442}(D),10948.08), \\ (B_{223}(D),11296.38),(B_{121}(D),11534.08),(B_{543}(D),11501.9), \\ (B_{132}(D),10948.08),(B_{4423}(D),0),(B_{1432}(D),0) \end{Bmatrix}$$ and

$U_4 = \{\} \quad i = 5$ .

**2.** *i=4* and $U_4$ is empty now*.*

**3.** *i=5*, and $U_5$ is not constructed . Go to step 12 of the algorithm.

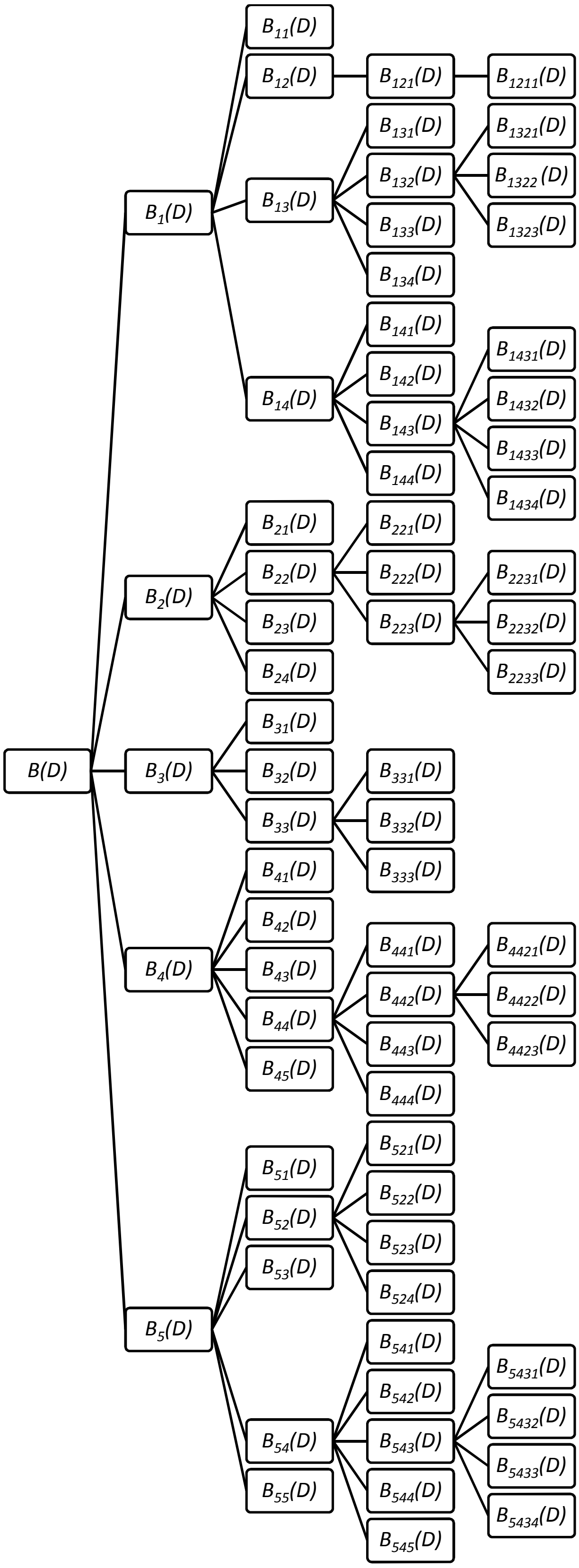



*Figure 4.4. The tree structure of the solutions of the example*

**4.** Find the biggest value of *π(D)* in *POS*, and show the corresponding $D^*$ and basis for that.

$B_{12}(D^*) \qquad \pi(D^*) = 11534.08$

$(D^*{}_{11}, D^*{}_{21}, D^*{}_{12}, D^*{}_{22}, D^*{}_{13}, D^*{}_{23}, D^*{}_{14}, D^*{}_{24}) =$
$(46.2, 52.5, 46.08, 51.6, 47.5, 55, 41.5, 50.1)$

**5.** Calculate the optimal price for each product by using the price function, $P_j(D_j)$. Show the values of x, y, z variables corresponding to the optimal basis.

$(p^*{}_{11}, p^*{}_{21}, p^*{}_{12}, p^*{}_{22}, p^*{}_{13}, p^*{}_{23}, p^*{}_{14}, p^*{}_{24}) =$
$(20.76, 19.5, 20.75, 19.68, 20.5, 19, 21.7, 19.98)$

$(x^*{}_{11}, x^*{}_{12}, x^*{}_{13}, x^*{}_{14}, x^*{}_{15}, x^*{}_{16}, x^*{}_{17}, x^*{}_{18}, x^*{}_{19}, x^*{}_{110}, x^*{}_{111}, x^*{}_{112}) =$
$(27.75, 18.50, 46.25, 50.69, 46.08, 41.47, 18.99, 33.24, 18.99, 20.78, 33.25, 49.88)$

$(x^*{}_{21}, x^*{}_{22}, x^*{}_{23}, x^*{}_{24}, x^*{}_{25}, x^*{}_{26}, x^*{}_{27}, x^*{}_{28}, x^*{}_{29}, x^*{}_{210}, x^*{}_{211}, x^*{}_{212}) =$
$(52.50, 53.41, 53.74, 49.30, 51.86, 51.66, 54.99, 54.99, 54.99, 50.11, 50.11, 50.11)$

We also formulated the above example by the model developed in Section 2.3. For the computational part, we utilised the existing optimisation packages with the capability of dealing with the nonlinear objective functions. We use the library subroutine 'NLPSolve' of Maple in which there is a method option to select the proper one for solving problems, such as Sequential Quadratic Programming (SQP). The solution of the numerical example with this approach is exactly the same as the solution obtained by the algorithm developed in Section 4.3.3. As a result, with this small example we show that the developed algorithm finds the same optimal solution as the other optimisation packages do. As discussed earlier, for the bigger size problems, the commercial packages may even fail to get a feasible solution, whereas with the presented algorithm we have the ease of solving a series of much smaller nonlinear problems over a set of linear constraints at each node of the search tree structure.

## 4.4. Conclusion

In this Chapter we have presented a mathematical programming model for determining the optimal production and pricing policy for a finite time horizon multiproduct production system with capacity constraints. Our model allows for backorders. Demand for each product is deterministic and dependent on its price, and the production set up cost is negligible. The mathematical programming formulation developed has a nonlinear objective function and some nonlinear constraints. This poses computational difficulties in large scale applications. To address this difficulty, we present a solution strategy that solves the nonlinear programming problem only under linear constraints, although it keeps the nonlinear constraints feasibility and can be effectively solved with an iterative two stage algorithm. The first stage of the algorithm finds the value of a Linear Programming's objective function in terms of the main problems' decision variables and in the second stage of the algorithm a Nonlinear Programming problem is solved subject to a linear constraint set. A detailed numerical example illustrates our solution strategy. The results of this chapter have been submitted for publication in June 2010.

# Chapter 5

# Fixed Pricing and Production Planning Under Uncertainty

What we have considered in previous Chapters deals only with deterministic demand. In this Chapter we incorporate the uncertainty of demand into our problem.

Although numerous models have been developed to solve deterministic joint pricing and production planning problems of multiple products, little work has been done on multi-product systems over a multi-period horizon under uncertainty.

E.Adida and G.Perakis (2006) use robust optimisation and fluid dynamic models to study a make-to-stock manufacturing system with uncertain demand. They show that the robust formulation is of the same order of complexity as the nominal (deterministic) problem and demonstrate how to adapt the nominal solution algorithm to the robust problem.

Generally in the case of uncertainty, it has been assumed that the demand has some known portion based on price (e.g., linear demand curve), with an additional stochastic element. However, in this chapter we deal with problems in which the demand/price relationship is uncertain with some known probabilities. In other words, we consider a discrete set of scenarios for the relationship between the chosen price and induced market demand. To our knowledge, this is the first time that the robust optimisation approach has been used in the case of discrete time production planning and pricing. The purpose of this Chapter is to develop a robust optimisation (RO) model to determine the optimal production planning and constant pricing of a manufacturing system with multiple products over a multiple period horizon to maximize the total profit which consists of sales revenue, production and inventory holding costs under demand/price uncertainty.

This Chapter is organized as follow: Section 5.1 briefly reviews the robust optimisation approach and its formulation in the case of a Linear Programming problem. Section 5.2 presents the deterministic model for the problem of joint pricing and production planning. It also proposes a robust optimisation model for the uncertain demand/price function. Section 5.3 reviews the existing solution methods developed for nonlinear programming problems. Section 5.4 illustrates our model and its solution with two practical size examples. The importance of this Chapter is the novel use of robust optimisation in the discrete case of joint pricing and production planning.

## 5.1. Robust optimisation approach

Our work is based on the robust optimisation tools developed by Mulvey, Vanderbei and Zenios (1995) which incorporates a goal programming structure with a set of scenarios involving stochastic inputs.

They are dealing with optimisation models that have two distinct components: a *structural* component that is fixed and free of any noise in its input data, and a *control* component that is subject to noisy input data. To define the appropriate model they introduce two sets of variables:

$x \in R^{n_1}$: the vector of decision variables whose optimal value is not conditioned on the realisation of the uncertain parameters. They are the *design* variables. Variables in this set cannot be adjusted once a specific realisation of the data is observed.

$y \in R^{n_2}$: the vector of *control* decision variables that are subjected to adjustment once the uncertain parameters are observed. Their optimal value depends both on the realisation of uncertain parameters, and on the optimal value of the design variables.

Their optimisation model has the following structure:

**Linear Programming problem**

$$Minimise \quad c^T x + \ d^T y \tag{5.1}$$

$$subject\ to \quad A\,x = b, \tag{5.2}$$

$$Bx + Cy = e, \tag{5.3}$$

$$x \in R^{n_1}\ ,\ y\ \in R^{n_2} \qquad and \qquad x, y \geq 0\ . \tag{5.4}$$

Constraint (5.2) denotes the structural constraints whose coefficients are fixed and free of noise. Constraint (5.3) denotes the control constraints. The coefficients of this constraint set are subject to noise. Constraint (5.4) expresses the non-negativity restrictions as well as pre-defined sets of variables.

To define the robust optimisation problem, a discrete set of scenarios $\Omega = \{1,2,3, \dots SN\}$ is introduced. With each scenario $s\ \in \Omega$ associate the set $\{d_s, B_s, C_s, e_s\}$ of realizations for the coefficients of the control constraints, and the probability of the scenario $P_s$, offcourse, $\sum_{s=1}^{SN} P_s = 1$. The optimal solution of the mathematical program (5.1)–(5.4) will be robust with respect to optimality if it remains "close" to optimal for any realization of the scenario $s\ \in \Omega$. It is then termed "*solution robust".* The solution is also robust with respect to feasibility if it remains "almost" feasible for any realization of $s$. It is then termed *"model robust".*

Because it is unlikely that any solution to problem (5.1)–(5.4) will remain both feasible and optimal for all scenario indices $s \in \Omega$, a model is needed that will allow us to measure the tradeoff between the solution and model robustness. The robust optimisation model proposed by Mulvey et al. (1995) formalises a way to measure this trade-off.

Let $\{y_1, y_2, \dots, y_{SN}\}$ be a set of control variables for each scenario $s \in \Omega$ and $\{z_1, z_2, \dots, z_{SN}\}$ a set of error vectors that measure the infeasibility allowed in the control constraints under scenario $s$. Consider now the following formulation of the robust optimisation model.

## Model ROBUST

$$Minimise \quad \sigma(\, x, y_1, \dots, y_{SN}) + \omega\rho(z_1, \dots, z_{SN}) \tag{5.5}$$

$$subject\ to \quad A\,x = b, \tag{5.6}$$

$$B_s x + C_s y_s + z_s = e_s, for\ all\ s \in \Omega \tag{5.7}$$

$$x \geq 0\,, y_s \geq 0, \text{ for all } s \in \Omega. \tag{5.8}$$

With multiple scenarios, the objective function $\varepsilon = c^T\,x + \; d^T\,y$ becomes a random variable taking the value $\varepsilon_s = c^T\,x + \; d^T{}_s\,y_s$, with probability $P_s$. Hence, there is no longer a single choice for an aggregate objective.

The first term of the objective function (5.5) measures optimality robustness. The goal programming weight $\omega$ is used to derive a spectrum of values that trade-off solution for model robustness.

For measuring the optimality we could use the mean value

$$\sigma(.) = \textstyle\sum_{s\,\in\Omega} P_s \varepsilon_s, \tag{5.9}$$

which is the function used in stochastic linear programming formulations. In a worst-case analysis the model minimizes the maximum value, and the first term of the objective function is defined by

$$\sigma(.) = max_{\ s \in \Omega}\ \varepsilon_s. \qquad (5.10)$$

Both of these choices are special cases of Robust Optimisation (RO).

The second term in the objective function $\rho(z_1, \dots, z_{SN})$ is a feasibility penalty function. It is used to penalize violations of the control constraints under some of the scenarios. The above model takes a multi-criteria objective form.

The specific choice of the penalty function is problem dependent, and also has implications for the accompanying solution algorithm. There are two alternative penalty functions considered:

$$\rho(z_1, \dots, z_{SN}) = \sum\nolimits_{s \in \Omega} P_s\, z_s^T\ z_s\ .$$

This quadratic penalty function is applicable to equality constrained problems.

$$\rho(z_1, \dots, z_{SN}) = \sum\nolimits_{s \in \Omega} P_s \max\{0, z_s\}\ .$$

This exact penalty function applies to inequality control constraints when only positive violations are of interest.

## 5.2. Joint pricing and production planning of multiple products over a multi-period horizon

In this Section we consider a multiproduct capacitated setting and introduce a demand-based model where the demand is a function of the price. There is an assumption that the production setup costs are negligible. A key part of the model is that the uncertain price/demand function is chosen from a discrete set of scenarios. As a result of this, the problem becomes a non linear programming problem with the nonlinearities only in the objective function. We develop a robust optimization model for this problem that considers the optimality and feasibility of all scenarios. The robust solution is obtained by solving a series of nonlinear programming problems. We illustrate our methodology with detailed numerical examples in the following sections.

### 5.2.1. Notation and model variables

In this part we bring the notation and terminology used by Gilbert (2000).

**Decision variables:**

$p_j$ : the price of product $j$; $j=1,2,\ldots,n$

$P$: the price vector

$D_j$ : the induced demand intensity for product $j$, which is a function of its price

$x_{jt}$ : the amount of product $j$ produced in period $t$; $j=1,2,\ldots,n$ and $t=1,2,\ldots,T$

$y_{jt}$ : the amount of product $j$ held in inventory at the end of period $t$

$X$ : the $n \times T$ production matrix

$Y$ : the $n \times T$ inventory matrix

**Parameters and constants:**

$c_j$ : the production cost of one unit of item $j$; $j=1,2,\ldots,n$

$h_j$ : the holding cost of one unit of item $j$ in inventory for one period

$K_t$ : the total amount of available capacity in period $t$

$\beta_{jt}$ : the seasonality parameter of item $j$ in period $t$

**Functions:**

$D_j(p)$ : the relationship between price and induced demand of $j$, for example: $D_j(p)= a_j- b_j.p_j$

## 5.2.2. Single scenario optimisation model (deterministic case)

We formulate the problem of jointly determining the price and production plan of multiple products over a multi-period horizon as follows.

**Objective**

The main components of the objective function in the problem of joint pricing and production planning include sales revenue, production cost and holding inventory cost. So, to consider the multi-product problem stated earlier as an aim of this work, the objective function is given as follows:

$$Max_{P,X,Y\geq 0}\{\pi(P,X,Y) = \sum_{j=1}^{n}\sum_{t=1}^{T} p_j . D_j(p) . \beta_{jt} - \sum_{j=1}^{n}\sum_{t=1}^{T} c_j . x_{jt} - \sum_{j=1}^{n}\sum_{t=1}^{T} h_j . y_{jt}\}. \quad (5.11)$$

The first term in (5.11) is the sales revenue as a product of the chosen price and the induced demand brought by the chosen price. The second and third terms are production and inventory holding costs respectively.

(5.11) can be written as:

$$Min_{P,X,Y\geq 0}\{-\pi(P,X,Y) = -\sum_{j=1}^{n}\sum_{t=1}^{T} p_j . D_j(p) . \beta_{jt} + \sum_{j=1}^{n}\sum_{t=1}^{T} c_j . x_{jt} + \sum_{j=1}^{n}\sum_{t=1}^{T} h_j . y_{jt}\}. \quad (5.12)$$

**Constraints**

$$\sum_{j=1}^{n}\sum_{t=1}^{T} \beta_{jt} . D_j(p) \leq \sum_{t=1}^{T} K_t \quad (5.13)$$

$$x_{jt} + y_{jt-1} - y_{jt} = D_j(p) . \beta_{jt} , for\ j = 1, \ldots n\ \ and\ t = 1, \ldots , T \quad (5.14)$$

$$\sum_{j=1}^{n} x_{jt} \leq K_t , for\ \ t = 1, \ldots , T \quad (5.15)$$

$$x_{jt}\ , y_{jt}\ , p_j \geq 0 , for\ j = 1, \ldots n\ \ and\ t = 1, \ldots , T. \quad (5.16)$$

Constraint (5.13) ensures that only demand intensity vectors which result in a feasible solution have been considered. Constraint (5.14) is a set of flow balance equations that ensure that all of the induced demand is satisfied. Constraint (5.15) ensures that there is an adequate amount of capacity in period *t* to produce all *n* items based on the plan. Inequality (5.16) expresses the non-negativity restrictions. We have a mathematical programming problem with a nonlinear objective and linear constraints.

### 5.2.3. Robust optimisation model for multiple scenarios

We assume that in the case of uncertainty, the price/induced demand function, $D_j(p)$, is not known in advance. Instead there is a discrete set of scenarios by which the relationship between price and induced demand is defined with a known probability. To find out the effect of uncertainty on the joint pricing and production planning, we need to redefine the objective function and some constraints, which are subject to the uncertainty, of the above single scenario model.

First, we redefine the objective function in (5.12) under each scenario, which consists of revenue, production cost and inventory holding cost.

$$RV^s\ (revenue) = \sum_{j=1}^{n} \sum_{t=1}^{T} p_j . D_j^s(p). \beta_{jt}\ , \qquad (5.17)$$

$$PC\ (production\ cost) = \sum_{j=1}^{n} \sum_{t=1}^{T} c_j . x_{jt}\ \ , and \qquad (5.18)$$

$$IC^s\ (inventory\ cost) = \sum_{j=1}^{n} \sum_{t=1}^{T} h_j . y_{jt}^s\ . \qquad (5.19)$$

In the above $D_j^s(p)$ is the price/induced demand function of product *j* and $y_{jt}^s$ is the inventory of product *j* at the end of period $t$ under scenario $s \in \Omega = \{1,2,3, \dots SN\}$ which happens with a probability of *Pr(s)*.The objective function for the joint pricing and production planning problem with noisy data comes to a random variable with the probability of each scenario, which is formulated as follows:

$$Min_{P,X,Y^s \geq 0} \ \sum_{j=1}^{n} \sum_{t=1}^{T} c_j . x_{jt} + \sum_{j=1}^{n} \sum_{t=1}^{T} h_j . y_{jt}^s - \sum_{j=1}^{n} \sum_{t=1}^{T} p_j . D_j^s(p) . \beta_{jt} \ . \tag{5.20}$$

Some might suppose that the expected objective function can be considered to cover all possible scenarios. But the optimal solution of the expected objective function is not likely to be optimal for all scenarios.

$$\begin{aligned} Min_{P,X,Y^s \geq 0} \ & \sum_{s \in \Omega} Pr(s) . \Big( \sum_{j=1}^{n} \sum_{t=1}^{T} c_j . x_{jt} + \sum_{j=1}^{n} \sum_{t=1}^{T} h_j . y_{jt}^s \\ & \qquad - \sum_{j=1}^{n} \sum_{t=1}^{T} p_j . D_j^s(p) . \beta_{jt} \Big) \\ & = \sum_{s \in \Omega} Pr(s) . (PC + IC^s - RV^s) \end{aligned} \tag{5.21}$$

Hence, the idea of adding a weight of the variance of the expected solution to the objective function causes choosing such solutions which are close to optimal for all scenarios. However, if we put a zero weight for the variance of the expected solution in the objective function, then we would have the general form of stochastic programming problem. On the other hand, in the case of uncertainty, the matter of feasibility should be taken into account as well as the optimality. To find a solution which remains almost feasible under all scenarios, we penalise the violation of feasibility of each constraint subject to uncertainty.

It is worthwhile to mention that mostly the robust optimisation approach has been used in linear programming problems as developed by Mulvey et al. (1995), but here, we develop a nonlinear programming model with the use of the RO approach. Now, the robust optimisation model for our targeted problem can be formally expressed as:

$$\begin{aligned} Min_{P,X,Y^s \geq 0} \ & \sum_{s \in \Omega} Pr(s) . (PC + IC^s - RV^s) \\ & + \lambda \sum_{s \in \Omega} Pr(s) . \Big[ (PC + IC^s - RV^s) \\ & \qquad - \sum_{s^\circ \in \Omega} Pr(s^\circ) . \big( PC + IC^{s^\circ} - RV^{s^\circ} \big) \Big]^2 \\ & + \omega \sum_{s \in \Omega} Pr(s) . \Big[ z^s + \sum_{j=1}^{n} \sum_{t=1}^{T} z_{jt}^s \Big] \end{aligned} \tag{5.22}$$

subject to:

$$\sum_{j=1}^{n} x_{jt} \leq K_t \ , \ for \ \ t = 1, \dots, T \tag{5.23}$$

$$\sum_{j=1}^{n} \sum_{t=1}^{T} \beta_{jt} \, . D_j^s(p) - z^s \leq \sum_{t=1}^{T} K_t \quad , for \ all \ s \in \Omega \tag{5.24}$$

$$x_{jt} + y_{jt-1}^s - y_{jt}^s + z_{jt}^s = D_j^s(p). \beta_{jt} \, , for \ j = 1, \dots n \ ; \ t = 1, \dots, T \ and \ all \ s \in \Omega, \tag{5.25}$$

$$x_{jt}, y_{jt}^s, p_j, z^s, z_{jt}^s \geq 0 \, . for \ j = 1, \dots n \ ; \ t = 1, \dots, T \ and \ all \ s \in \Omega \tag{5.26}$$

Note that $z_{jt}^s$ is the under-fulfillment of demand of product *j* in period *t* under scenario *s.* Also $z^s$ is the total under-fulfilment of demand of all products over the total planning horizon under scenario *s.*

The first and second terms in the objective function (5.22) are mean and variance of the objective function respectively, which measure the solution robustness. The third term in (5.22) is set to measure the model's robustness with respect to infeasibility associated with control constraints (5.24) and (5.25) under scenario *s*.

Constraints (5.23) ensure that there is an adequate amount of capacity in period *t* to produce all *n* items based on the plan, which are free of noise. Constraints (5.24) consider only demand intensity vectors which result to a feasible solution for the total production capacity available over the planning horizon. Because these are control constraints which are subject to uncertainty, $z^s$ measures the amount of infeasibility under scenario *s*. Constraints (5.25) are a set of flow balance equations that ensure that all of the induced demand is satisfied. Due to the noisy data, a new variable, $z_{jt}^s$ , is introduced for each product of type *j* in period *t* to keep the track of feasibility under scenario *s*. Finally, constraints (5.26) are simply the non-negativity requirements.

## 5.3. Solution Methods

Given the capacity limitations and uncertainty in the problems parameters, the firm must decide upon production quantities, inventory levels for each item as well as a constant price at which it commits to sell the products over the total planning horizon.

As already been noted, corresponding to each specific $\lambda$ and $\omega$, which define the optimality and feasibility preferences, the problem (5.22)-(5.26) comes to an optimisation problem with nonlinear objective function and linear constraints. There is a vast literature on such problems. The book by Luenberger (2003) presents different methods designed to solve a Nonlinear Programming problem which has $n$ variables and $m$ constraints. Methods devised for solving this problem that work in spaces of dimension $n$-$m$, $n$, $m$, or $n$+$m$ are Primal Methods, Penalty and Barrier Methods, Dual and Cutting Plane Methods and Lagrange Methods, respectively.

Primal methods work on the original problem directly by searching through the feasible region, which has dimension $n$-$m$, for the optimal solution. Each point in the process is feasible and the value of the objective function constantly improves. Penalty and Barrier methods approximate constrained optimization problems by unconstrained problems with adding a term to the objective function. In the case of penalty methods the term prescribes a high cost for violation of the constraints, and in the case of barrier methods the term favours the points interior to the feasible region over those near the boundary. Dual methods are based on the viewpoint that it is the Lagrange multipliers which are the fundamental unknowns associated with a constrained problem. Once these multipliers are known, the determination of the solution point is simple. Dual methods do not attack the original constrained problem directly but instead attack an alternate problem, the dual problem, whose unknowns are the Lagrange multipliers of the first problem. Cutting plane algorithms develop a series of ever-improving approximating linear programs, whose solutions converge to the solution of the original problem. Lagrange methods directly solve the Lagrange first-order necessary conditions. The set of necessary conditions is a system of *n+m* equations in the *n+m* unknowns.

In the computation part of this work we have utilised the existing optimisation packages with the capability of dealing with the nonlinear objective functions. We use the library subroutine 'NLPSolve' of Maple in which there is a method option to select the proper one for solving the specific problem, such as Quadratic Interpolation, Branch-and Bound, Modified Newton, Nonlinear Simplex, Preconditioned Conjugate Gradient and Sequential Quadratic Programming (SQP). According to the described criteria for each method, we have selected SQP to optimize each NLP problem.

Our strategy to find the optimal pricing and production planning under uncertainty is to solve a sequence of the above nonlinear programming problems for a range of $\lambda$ and $\omega$ . We first choose a fixed small value (e.g. 0.01) of $\lambda$ and a considerably vast range of $\omega$ (e.g. 0-300, with suitable increments) and find the optimal solution of each specific problem within the selected range. We draw three different plots of all specific problems as follows, the expected profit, the solutions standard deviation (as a measurement of the optimality) and the demand under fulfilment (as a measurement of the feasibility). Next, if the expected profit doesn't level out within the chosen range of $\omega$, we extend the range to observe a leveled out expected profit. By increasing the value of $\lambda$ , we solve the sequence of nonlinear problems again within the same modified range of $\omega$. Comparing the resulted plots of the new $\lambda$ and the previous one, and bearing in the mind the decision makers preferences, the next value of $\lambda$ may be selected to continue the computation. As would be expected, the robust zone of any specific case of joint pricing and production planning is highly dependent on the resulting plots to compare the expected profit with the optimality and feasibility measures.

In the next Section we illustrate this RO approach and our strategy to find the robust solution for a detailed small example. We further demonstrate the ability to handle a large, more realistic case.

## 5.4. Numerical Example

In order to make the model more clear, two different cases are presented in this Section. First, we start modelling and solving a smaller example with *n=2* products , *T =6* periods and $\Omega = \{1,2\}$. The second example consists of *n=10* products , *T =12* periods and $\Omega = \{1,2,3\}$.

### 5.4.1. Two products, Six Periods and Two Scenarios

The parameters for the example are as shown in Table 5.1.

*Table 5.1. Parameters of the example.*

| ***Product*** | ***Scenario*** | $D_j^s(p)$ | $h_j$ | $c_j$ | $\beta_{j1}$ | $\beta_{j2}$ | $\beta_{j3}$ | $\beta_{j4}$ | $\beta_{j5}$ | $\beta_{j6}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| ***j=1*** | ***Pr(s=1)=0.8*** | *150-5* $p_1$ | 6 | 16 | 0.6 | 0.5 | 0.2 | 3 | 1.5 | 0.2 |
| | ***Pr(s=2)=0.2*** | *150-4* $p_1$ | | | | | | | | |
| ***j=2*** | ***Pr(s=1)=0.8*** | *150-5* $p_2$ | 2.5 | 11 | 1 | 1 | 1 | 1 | 1 | 1 |
| | ***Pr(s=2)=0.2*** | *150-4* $p_2$ | | | | | | | | |

We assume a fixed production capacity, $K_t$ =l40, for all periods in the planning horizon.

Here we bring the problem formulation based on the proposed RO approach as follows:

$$RV^1 = 6\, p_1\, (150 - 5\, p_1) + 6\, p_2\, (150 - 5\, p_2)$$
$$= 900\, p_1 + 900\, p_2 - 30\, p_1^2 - 30\, p_2^2$$
$$RV^2 = 6\, p_1\, (150 - 4\, p_1) + 6\, p_2\, (150 - 4\, p_2)$$
$$= 900\, p_1 + 900\, p_2 - 24\, p_1^2 - 24\, p_2^2$$
$$PC = 16\, (x_{11} + x_{12} + x_{13} + x_{14} + x_{15} + x_{16})$$
$$+ 11\, (x_{21} + x_{22} + x_{23} + x_{24} + x_{25} + x_{26})$$
$$IC^1 = 6\, (y_{11}^1 + y_{12}^1 + y_{13}^1 + y_{14}^1 + y_{15}^1) + 2.5\, (y_{21}^1 + y_{22}^1 + y_{23}^1 + y_{24}^1 + y_{25}^1)$$
$$IC^2 = 6\, (y_{11}^2 + y_{12}^2 + y_{13}^2 + y_{14}^2 + y_{15}^2) + 2.5\, (y_{21}^2 + y_{22}^2 + y_{23}^2 + y_{24}^2 + y_{25}^2)$$

The problem is:

$$Min \qquad PC + 0.8(IC^1 - RV^1) + 0.2(IC^2 - RV^2)$$
$$+ 0.8\, \lambda\, [0.2(IC^1 - RV^1) - 0.2(IC^2 - RV^2)]^2$$
$$+ 0.2\, \lambda\, [0.8(IC^2 - RV^2) - 0.8(IC^1 - RV^1)]^2$$
$$+ 0.8\, \omega\, [z^1 + z_{11}^1 + z_{12}^1 + z_{13}^1 + z_{14}^1 + z_{15}^1 + z_{16}^1 + z_{21}^1 + z_{22}^1$$
$$+ z_{23}^1 + z_{24}^1 + z_{25}^1 + z_{26}^1]$$
$$+ 0.2\, \omega\, [z^2 + z_{11}^2 + z_{12}^2 + z_{13}^2 + z_{14}^2 + z_{15}^2 + z_{16}^2 + z_{21}^2 + z_{22}^2$$
$$+ z_{23}^2 + z_{24}^2 + z_{25}^2 + z_{26}^2].$$

*subject to:*

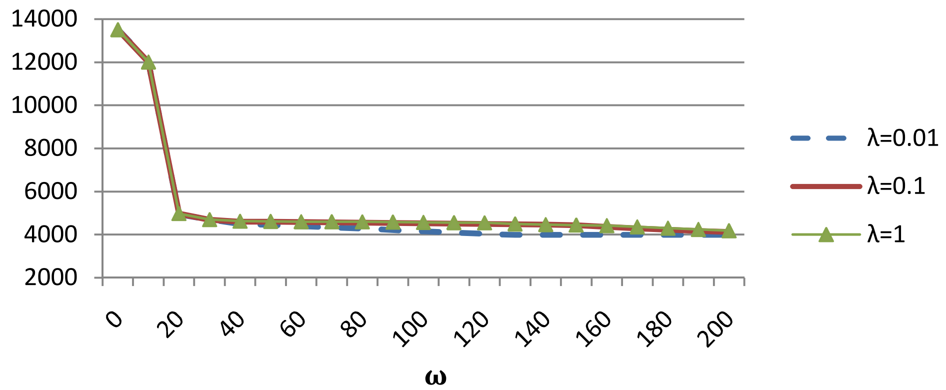
$$x_{1t} + x_{2t} \leq 140, for\ \ t = 1, \dots, 6$$
$$6\, (150 - 5\, p_1) + 6\, (150 - 5\, p_2) - z^1 \leq 840$$
$$6\, (150 - 4\, p_1) + 6\, (150 - 4\, p_2) - z^2 \leq 840$$
$$x_{jt} + y_{jt-1}^s - y_{jt}^s + z_{jt}^s = D_j^s(p).\beta_{jt}\ for\ j = 1,2\ ;\ t = 1, \dots, 6\ and\ all\ s \in \{1,2\},$$
$$x_{jt}, y_{jt}^s, p_j, z^s, z_{jt}^s \geq 0, for\ j = 1,2\ ;\ t = 1, \dots, 6\ and\ all\ s \in \{1,2\}.$$

The Nonlinear Programming problem has been solved for a range of λ and ω. The optimized value of the decision variables can be obtained for each specific problem with a particular λ and ω.

Based on the decision maker's preferences the violation of the optimality and feasibility can be penalized by choosing the appropriate value for λ and ω respectively. The higher value of λ results in a solution with less standard deviation from the expected one and a big ω brings more feasibility to all control constraints under each scenario. Hence, a single decision cannot be made instantly for this type of problem with uncertainty; instead there should be a reasonable discussion revealing the importance of the optimality and feasibility for each case.

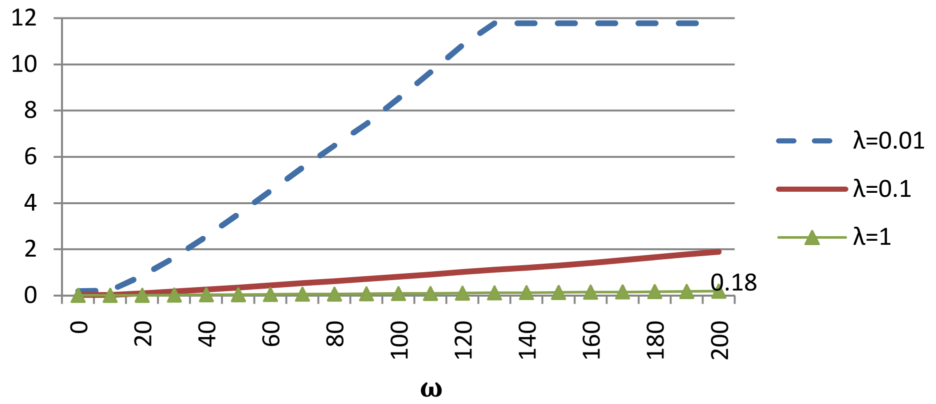
Now we bring the result of the specific problem for a chosen range of ω (0-200) and some fixed value of λ (0.01, 0.1, 1).

*Figure 5.1. The expected total profit*



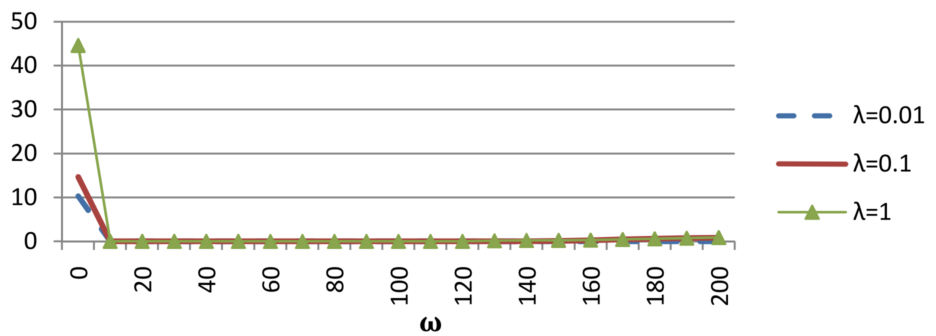

As can be seen, the expected profit drops dramatically when the value of ω increases by 20. Since then, by increasing the value of ω, the change of expected profit is not considerable. When the value of λ rises from 0.1 to 1, we can say that the total profit remains very similar. As a result, choosing the preferred λ depends on the other aspects of the decision making, which are optimality and infeasibility measurements illustrated in the following figures. Figure 5.2. illustrates the solution's standard deviation as a percentage of the expected total profit, which is actually an indicator of the optimality. Figure 5.3. shows the total demand under-fulfilment as a percentage of the total expected demand, which measures the feasibility.

*Figure 5.2. The solution's standard deviation*




As should be expected, we are interested in the smaller values on this figure, because they give a better solution regarding the optimality. Like the previous figure, by increasing the value of λ from 0.1 to 1, the solution's standard deviation does not change significantly. So, to choose the preferred λ we need the result of the feasibility measurement to be able to summarize the robust decision making.

*Figure 5.3. The total demand under-fulfilment*




By increasing the value of ω from zero to 10, the percentage of demand under-fulfillment drops sharply. Also by raising λ, the amount of unsatisfied demand will be increased too. But, because at any nonzero rate of ω the under-fulfillment is less than five percent, we do not have too much concern about choosing larger values for λ which brings more optimality. In other words, for finding the robust answer to this specific example, we just consider first and second figures as the mean and standard deviation of the solution respectively.

By finding the optimal solution for λ=10 over the same range of ω and comparison to the above plots, it can be seen that almost all the results remain similar. As a result, λ=1 and $\omega = 20$ to $30$ provides a reasonable zone to find the robust solution of the problem. As an example we present the output within the robust zone for λ=1 and $\omega = 20$.

The total expected profit: $4971.1$

The standard deviation: $\% \, 0.0084$

The total under-fulfillment: $0$

$x_{11} = 23.0844$ $x_{12} = 19.2370$ $x_{13} = 7.6948$ $x_{14} = 7.6948$

$x_{15} = 57.7110$ $x_{16} = 63.6037$

$x_{21} = x_{22} = x_{23} = x_{24} = x_{25} = x_{26} = 51.7045$

$p_1 = 22.3051$ $p_2 = 19.6590$

## 5.4.2. Ten products, Twelve Periods and Three Scenarios

We now discuss a realistic size problem in manufacturing pricing, where the firm is not flexible to change the price list frequently and usually has long-term contracts with Original Equipment Manufacturers (OEMs). It is reasonable to expect that in a real situation the number of product classes, which do not have cross price dependency in their demand function is not too large. So the assumption of *n=10* can cover a large number of applications. Besides, by considering *T=12* we are catering for a whole year of planning on a monthly basis, which is logical for constant pricing. Our three scenarios present good, moderate and weak market situations. For such a realistic size example, Maple is efficient.

The parameters for this example are as shown in Table 5.2.

*Table 5.2. Parameters of the example.*

| **Product** | $h_j$ | $c_j$ | $\beta_{jt}$ | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| ***j=1*** | 2 | 6 | 0.6 | 0.5 | 0.2 | 0.2 | 1.5 | 3 | 2 | 1 | 0.8 | 0.5 | 0.7 | 1 |
| ***j=2*** | 1 | 4 | 0.8 | 0.6 | 1 | 2 | 1.5 | 1.1 | 1 | 0.9 | 0.7 | 1.4 | 0.3 | 0.7 |
| ***j=3*** | 3 | 8 | 0.7 | 0.4 | 0.3 | 0.3 | 1.5 | 2.8 | 2 | 1 | 0.7 | 0.6 | 1 | 0.7 |
| ***j=4*** | 2 | 5 | 0.9 | 0.3 | 0.6 | 1.2 | 1.5 | 1.5 | 3 | 0.5 | 0.6 | 0.4 | 0.7 | 0.8 |
| ***j=5*** | 4 | 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| ***j=6*** | 1 | 7 | 2 | 1 | 0.8 | 0.5 | 0.7 | 1 | 0.6 | 0.5 | 0.2 | 0.2 | 1.5 | 3 |
| ***j=7*** | 4 | 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| ***j=8*** | 3 | 4 | 1.5 | 1.5 | 1.2 | 0.6 | 0.3 | 0.9 | 1 | 2 | 0.6 | 0.7 | 1 | 0.7 |
| ***j=9*** | 2 | 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| ***j=10*** | 1 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| ***Scenario*** | $D_j(p_j)$ |
|---|---|
| ***Pr(s=1)=0.7*** | *150-5* $p_j$ |
| ***Pr(s=2)=0.2*** | *150-4.5* $p_j$ |
| ***Pr(s=3)=0.1*** | *150-4* $p_j$ |

We assume a fixed production capacity, $K_t$ =I00, for all periods in the planning horizon. Similar to the previous example, the results for this specific problem include the expected profit, the standard deviation of the solution and the total demand under-fulfilment shown as follow:

*Figure 5.4. The expected total profit*

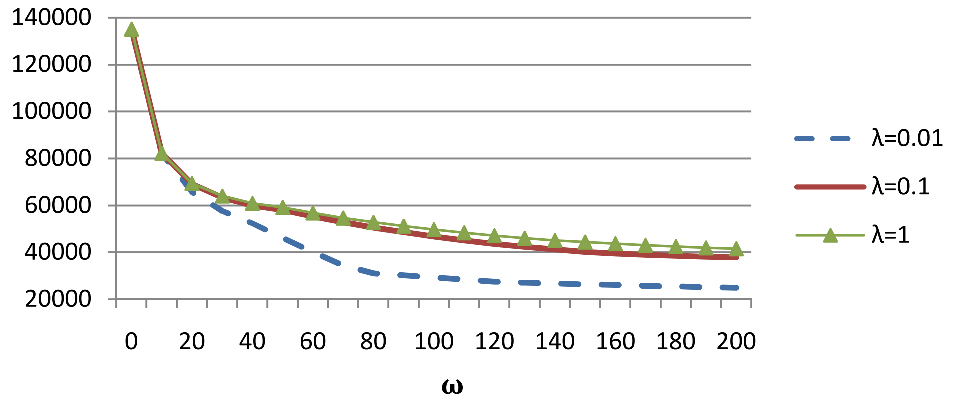



*Figure 5.5. The solution's standard deviation as a percentage of the expected total profit*

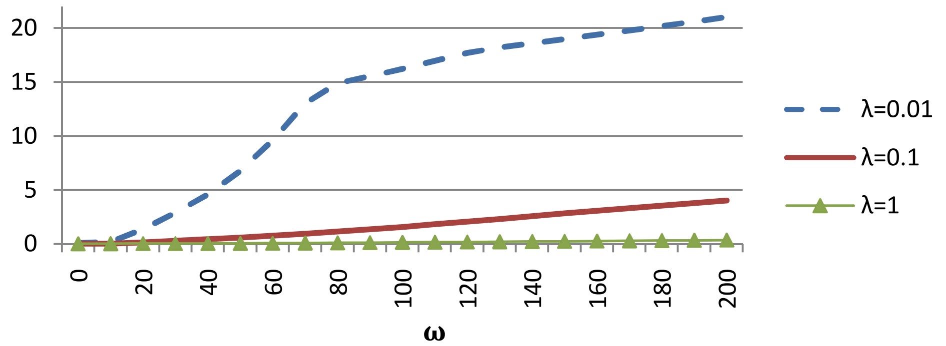
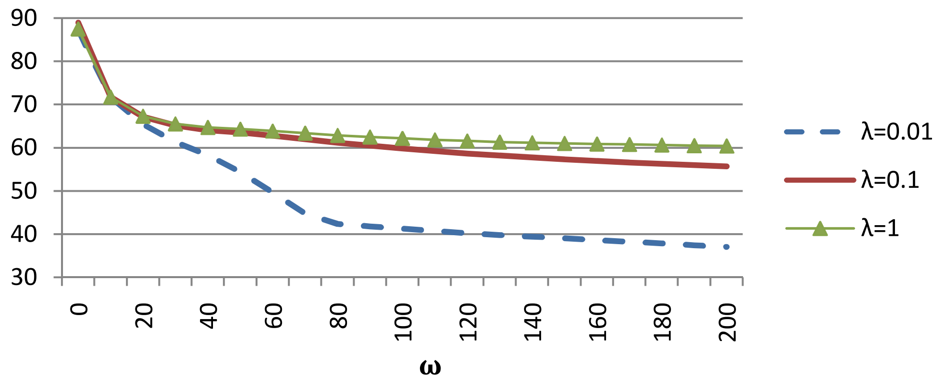



*Figure 5.6. The total demand under-fulfilment as a percentage of the expected total demand*




As can be seen, the expected profit in the case of λ=0.1 and λ=1 remains very similar. We need to consider the optimality and infeasibility measurements simultaneously to choose the robust zone preferred by the decision maker. If it is desired to have the demand satisfied as much as possible, we might choose λ=0.01 and $\omega = 80 \text{ to } 90$ which results in the smallest possible standard deviation and the biggest possible expected profit within the chosen range of λ and ω .On the other hand, if the decision maker is concerned more about the profit regardless of the unsatisfied demand, we might select a bigger value of λ (like 1) and a range of ω (like 20-30) after which the expected profit and infeasibility amount remain much similar. Thus, our methodology presents tools which assist the decision maker explore the range of possibilities.

## 5.5. Conclusion

In this Chapter we have presented a mathematical programming model for determining the optimal production and constant pricing policy for a finite time horizon multiproduct production system with capacity constraints and demand uncertainty. The production set up cost is negligible, and demand for each product is dependent on its price, but the price/demand function is uncertain. Our methodology makes use of Robust Optimisation ideas and our model can be effectively implemented utilizing existing computational packages (we use Maple). We illustrate with detailed numerical examples. The results of this Chapter are to appear in the Pacific Journal of Optimisation, 2011, in press.

# Chapter 6

# Conclusion and future work

In this thesis, we make several contributions to the literature of joint pricing and production planning. More specifically, this work develops and analyses several inventory/production models with pricing decisions of multiple products over a multiperiod horizon under capacity constraints. It extends some existing results in constant pricing and inventory/production systems and provides an approach to develop a tree search algorithm for the time-variant problem with and without backorders. This thesis also utilises a robust optimisation approach for handling the demand/price uncertainty induced from market.

We include the dynamic pricing decision into multiproduct inventory/production systems. We formulate the problem as a Non-Linear Programming Problem in both cases of with and without backorders and discuss the limitations of getting the optimal solution due to the price incorporation. In this thesis, we mainly address the reduction of the level of difficulty of the formulated optimisation problem and develop an efficient algorithm to solve it. These tackles include considering a special but practical case for the demand/price function, conversion of nonlinear constraints to linear and use of results for constant pricing.

We contribute to develop a solution strategy to solve the constant pricing with allowed backorders. The strategy constructs a tree search in breadth-first manner iteratively. A detailed example shows the ability of the proposed algorithm to be implemented in practical size problems.

We extend our work to the case where the price and other parameters of the problem are time-variant. By this extension, our aim is to tackle the formulated Non Linear Programming problem in the earlier part by dividing it into several sub-problems with convex set of constraints. Our solution Strategy proposes an iterative algorithm to construct a search tree structure to find the optimal solution more efficiently. The detailed algorithm is illustrated through a numerical example.

We also contribute in incorporating the demand/price uncertainty into the model. We make use of the robust optimisation approach to handle the existence of several scenarios in the market. Our strategy is to utilise existing computational packages (like Maple) for solving the problem. We illustrate our method with detailed numerical examples.

The outcomes of the research presented in this thesis have provided us with a better understanding of the joint pricing and production planning problem along with its complexities. However there are still areas that are open to further investigation and perhaps improvement.

One of the main goals of this thesis was to incorporate endogenous demand into the inventory/production systems. More clearly, we tried to see what the effect of pricing decisions would be on production plans due to the demand/price relationship. This was done through the assumption of zero price elasticity. There is still scope to improve this investigation. For example, one might consider product substitution or non-zero price elasticity.

Another area of the research that can be explored further is the consideration of strategic behaviour of customers. Whether this increase in complexity in the model improves the quality of practical coverage would be worth investigating.

# Refrences:


Adida E and G.Perakis, "A robust Optimisation Approach to Dynamic Pricing and Inventory Control with No Backorders*". Mathematical Programming*, **Ser. B 107,** 2006, 97-129.

Adida E, Perakis G, "A Nonlinear Continuous Time Optimal Control Model of Dynamic Pricing and Inventory Control with No Backorders". *Wiley InterScience*, 2007.

Arrow K. J ,"Studies in applied probability and management science", *Stanford University Press,* 1962.

Bassok Y, Anupindi R, and Akella R, "Single Period Multiproduct Inventory Models With Substitution". *Operations Research*, **47(4)**, 1999, 632-642.

Bernstein F and Federgruen A, "Pricing and replenishment strategies in a distribution system with competing retailers". *Operations Research*, **51(3)**, 2003, 409-426.

Biller S, Chan L.M, Simchi-Levi D, and Swann J*, "*Dynamic pricing and the direct-to-customer model in the automotive industry". *Electronic Commerce Research*, **5(2),** 2005, 309-334.

Birge J. R, Drogosz J, and Duenyas I, "Setting single-period optimal capacity levels and prices for substitutable products". *International Journal of Flexible Manufacturing Systems*, **10(4)**, 1998, 407-430.

Black Paul E., "greedy algorithm" in Dictionary of Algorithms and Data Structures ,*U.S. National Institute of Standards and Technology*, February 2005, webpage: NIST-greedyalgo.

Caccetta L and Mardaneh E, “Joint pricing and production planning for fixed priced multiple products with backorders”. *Journal of Industrial and Management Optimisation*, **6(1)**, 2010, 123-147.

Caccetta L and Mardaneh E, “Joint pricing and production planning of multi-period multi-product systems with uncertainty in demand”. *To be appeared in Pacific Journal of Optimisation,* 2010.

Chan L.M, Simchi-Levi D, and Swann J, “Dynamic pricing models for manufacturing with stochastic demand and discretionary sales”. *Manufacturing and Service Operations Management*, **8(2)**, 2006, 149-168.

Eliashberg J and Steinberg R, “Marketing-production joint decision making”. *Management Science in Marketing, Handbooks in Operations Research and Management Science. Elsevier Science, North Holland*, 1991.

Elmaghraby W, Gulcu A and Keskinocak P, “Designing Optimal Preannounced Markdowns in the Presence of Rational Customers with Multiunit Demands”, *Manufacturing & Service Operations Management*, **10(1)**, 2008, 126-148.

Gallego G and Van Ryzin G, “A multiproduct dynamic pricing problem and its applications to network yield management”. *Operations Research*, **45(1)**, 1997, 24-41.

Gilbert S. M, “Coordination of pricing and multi-period production for constant priced goods”. *European Journal of Operational Research*, **114(2)**, 1999, 330-337.

Gilbert S. M, “Coordination of pricing and multiple-period production across multiple constant priced goods”. *Management Science*, **46(12)**, 2000, 1602-1616.

Kachani S and Perakis G, “A fluid model of dynamic pricing and inventory management for make-to-stock manufacturing systems”. *Massachusetts Institute of Technology, Operations Research Center,* 2002.

Karakul M and Chan L. M. A, “Analytical and managerial implications of integrating product substitutability in the joint pricing and procurement problem”,*European Journal of Operational Research*, **116**, 2008, 179–204.

Kunreuther H and Schrage L, “Joint pricing and inventory decisions for constant priced items”. *Management Science*, **19(7)** , 1973, 732-738.

Luenberger G, “Linear and Nonlinear Programming*”,* 2003, 322-454.

Meyer R. A, “Risk-efficient monopoly pricing for multiproduct firms”. *Quarterly Journal of Economics*, **90(3)**, 1976, 461-474.

Morgan L. O, Daniels R. L and Kouvelis P, “Marketing/ manufacturing trade-offs in product line management”. *IIE Transactions*, **33(11)**, 2001, 949-962.

Mulvey J.M, Vanderbei R.J and Zenios S.A, “Robust Optimisation of Large-scale Systems*”*. *Operations Research*, **43(2)**, 1995, 264-281.

Neale R. S, Jorge L.R and Leopoldo Eduardo C.B, “Optimal Pricing and Production Master Planning in a Multiperiod Horizon Considering Capacity and Inventory Constraints”, *Mathematical Problems in Engineering,* Volume 2009, Article ID 932676, 15 pages.

Netessine S, “Dynamic pricing of inventory/capacity with infrequent price changes”, *European Journal of Operational Research*, **174** ,2006, 553-580.

Petruzzi N. C and Dada M, “Dynamic pricing and inventory control with learning”, *Naval Research Logistics*, **49(3),** 2002, 303-325.

Rajan A , Rakesh and Steinberg R, “Dynamic pricing and ordering decisions by a monopolist”. *Management Science*, **38(2)**, 1992, 240-262.

Russell S. J, Norvig P, “Artificial Intelligence: A Modern Approach”, *Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2, http://aima.cs.berkeley.edu/,* 2003, 111-114.

Simchi-Levi D, Wu S. D and Shen Z. J, “Modeling in the E-business Era”. *Handbook of Qualitative Supply Chain Analysis*, 2004, 335-392.

Smith S. A, Agrawal N, and McIntyre S. H, “A discrete optimization model for seasonal merchandise planning”. *Journal of Retailing*, **74(2)**, 1998, 193-221.

Streitfeld D, “On the web, price tags blur; what you pay could depend on who you are”. *The Washington Post*, Sept 27 2000.

Vizard M, Scannell E and Neel D, “Suppliers toy with dynamic pricing”, *InfoWorld*, May 11, 2001.

Yano C and Dobson G, “Profit-optimizing product line design, selection and pricing with manufacturing cost considerations”. *T.H. Ho and C. S. Tang, editors, Product Variety Management: Research Advances,* Kluwer, Norwell, MA , 1998, 145-176.

Zhao W, Zheng Y. S, “Optimal dynamic pricing for perishable assets with nonhomogeneous demand”, *Management Science*, **46(3)**, 2000, 375-388.

Zhu K , Thonemann U. W, “Coordination of pricing and inventory control across products”. *Naval Research Logistics*, **56(2)** ,2009, 175-190.

Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.