Roll Assortment Optimization in a Paper Mill: An Integer Programming Approach

Satyaveer Singh Chauhan,

Alain Martel

and

Sophie D'Amours

January 2006

Working Paper DT-2005-AM-6

(To appear in Computers and Operations Research)

Research Consortium on e-Business in the Forest Products Industry (FORAC),
Network Organization Technology Research Center (CENTOR),

Université Laval, Québec, G1K 7P4, Canada

Roll Assortment Optimization in a Paper Mill: An Integer Programming Approach

S. S. Chauhan, Alain Martel*[*] and Sophie D'Amour

FOR@C Research Consortium,
Network Organization Technology Research Center (CENTOR),
Université Laval, Sainte-Foy, Québec, G1K 7P4, Canada.

Abstract: Fine paper mills produce a variety of paper grades to satisfy demand for a large number of sheeted products. Huge reels of different paper grades are produced on a cyclical basis on paper machines. These reels are then cut into rolls of smaller size which are then either sold as such, or sheeted into finished products in converting plants. A huge number of roll sizes would be required to cut all finished products without trim loss and they cannot all be inventoried. An assortment of rolls is inventoried with the implication that the sheeting operations may yield trim loss. The selection of the assortment of roll sizes to stock and the assignment of these roll sizes to finished products have a significant impact on performances. This paper presents a model to decide the parent roll assortment and assignments to finished products based on these products demand processes, desired service levels, trim loss and inventory holding costs. Risk pooling economies made by assigning several finished products to a given roll size is a fundamental aspect of the problem. The overall model is a binary non-linear program. Two solution methods are developed: a branch and price algorithm based on column generation and a fast pricing heuristic, and a marginal cost heuristic. The two methods are tested on real data and also on randomly generated problem instances. The approach proposed was implemented by a large pulp and paper company.

Keywords: Assortment optimization, risk pooling, paper cutting, integer programming, column generation.

Acknowledgements: This project would not have been possible without the collaboration of FOR@C's partners and especially of NSERC and Domtar.

---

[*] Corresponding author e-mail: alain.martel@centor.ulaval.ca.

1. Introduction

Production planning in the fine paper industry is extremely difficult because a huge variety of finished paper products is demanded by customers. Generally, the fine paper making process involves the following two major stages:

1. Fixed width *reels* of a given paper grade, also referred to as *jumbo rolls*, are produced on paper machines in a paper mill. These reels are then cut into several *rolls* of smaller diameters and widths on a winder. Some of the rolls produced are sold directly to customers.

2. A significant part of the rolls produced in the paper mill are transformed into cut-sheet finished products on sheeters in a converting plant, which may generate some trim loss. The rolls sheeted into finished products are known as *parent rolls*. In order to respond quickly to customer demands, an inventory of parent rolls may be kept at the converting plant.

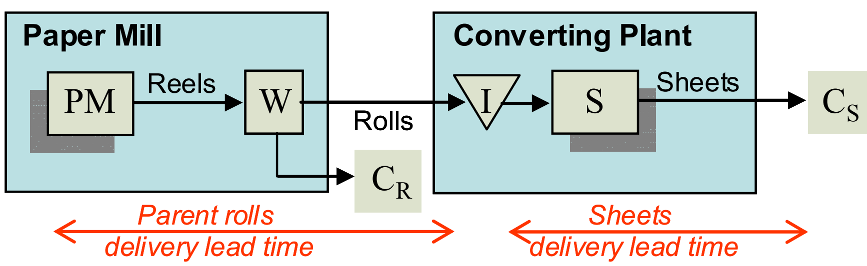
Note that the paper mill and the converting plant are not necessarily located on the same site.

In this work we concentrate on the fulfillment of demand for cut-sheet paper products such as copy, offset, designer and packaging paper. This problem is complex due to the huge number of sheet dimensions demanded and to the stochastic nature of demands. The objective is to improve yield and customer service levels at the lowest possible cost. Paper machines operate in production cycles (commonly called production campaigns) and the length of a cycle can be a week or two. During a production cycle a machine produces several paper grades in a given sequence but the production quantity for a grade is based on external and internal roll orders. Considering this, three different order penetration points can be adopted in fine paper supply chains (Lehtonen and Holmström, 1998; Hameri and Nikkola, 2001), namely: Make-to-order (MTO), Sheet-to-order (STO) and Make-to-stock (MTS).

Under a MTO strategy, cut-sheet customer orders for a given paper grade are grouped into predetermined reel cutting patterns and the needed reels are scheduled for production on the paper machines. With this approach no intermediate stock is required and the total delivery lead time is the sum of the paper machine production cycle, the paper cutting times and the transportation time. This approach is very cost effective as inventory costs are minimal and trim costs can be minimized using an appropriate two-phase cutting stock problem algorithm; but it is appropriate only if customers are prepared to wait. Integrated scheduling and cutting approaches for the problem are proposed by Keskinocak et al. (2002) and by Correira et al. (2004). Roll cutting models particularly suited for the paper industry are presented by Goulimis (1990), Sweeney and Haessler (1990) and Westerlund et al. (1998). The MTO strategy is most suitable in the case of deterministic demands, particularly if it takes the form of large orders, which is frequent for rolls but relatively rare for sheets (Hameri and Nikkola, 2001).

To address the randomness in cut-sheet demand, safety buffers must be created either by stocking reels, parent rolls, finished products or a combination. Since reels are too huge to stock, this option is not practical. When a parent roll stock is used as a buffer, the paper cutting process is decoupled and finished products are sheeted to order. In order to implement this STO strategy, an assortment of parent roll sizes to stock must be selected and a roll inventory management system must be implemented. When a finished product customer order arrives, adequate parent rolls are taken out of stock and cut to the required size on a sheeter. This generates trim loss and one of the rolls sheeted may not be completely used. The balance of the roll left is known as *overrun*. The advantage of this order penetration strategy is that the delivery lead time is reduced significantly: it now includes only the paper sheeting time and the transportation time, assuming enough stock is available in the parent roll buffer. On the other end, it increases the inventory holding costs and it may increase the trim loss cost. However, in a market where delivery lead times are an order winning criterion, this strategy may bring a competitive advantage (Lehtonen and Holmström, 1998).

A third option is a MTS strategy in which buffers are introduced at the finished products stage. This strategy leads to shorter delivery times and superior customer service, but it generates high inventory holding costs. Synchronized planning approaches for this strategy were proposed by Krichagina et al. (1998), for the stochastic demand case, and by Rizk et al. (2005), for the dynamic demand case. Since the unit inventory holding cost is higher for finished products than for parent rolls (because of the value added at the sheeting stage), and since the variance to mean ratio of the finished products demand is much higher than for the parent rolls demand (because of risk pooling), the inventory costs for this strategy can be much higher than for the sheet-to-order strategy. In fact, this strategy should be adopted only when rapid deliveries are required to compete, which is often the case for standardized high-volume cut-sheet paper products (Hameri and Nikkola, 2001). Clearly, in practice, a mix strategy is also possible.



PM: Paper machines W: Winder I: Parent rolls inventory S: Sheeters
C_R: Roll customers C_S: Sheet customers

Figure 1: Fine Paper Supply Chain under a Sheet-to-order Strategy

Despite its high potential, little work was done on the STO strategy. The fine paper supply chain obtained under this strategy is illustrated in Figure 1. As indicated earlier, in order to implement a STO strategy, the assortment of parent roll sizes to stock in the converting plant must be selected, the parent roll to use to cut each finished product on the sheeters must be specified and a parent roll inventory management system must be implemented. Several hundred finished products may exist and it is assumed that their demand follows independent stationary stochastic processes with known mean and variance. Given this, the demand process for a parent roll is the convolution of the demand of the products it is assigned to, and it is thus also stationary. It is well established in the inventory management literature (Silver *et al.*, 2003) that the inventory of stationary demand products can be managed with periodic review or continuous transaction inventory systems, and that product supply orders can be coordinated or not. All these inventory management systems give rise to two types of inventories, order cycle stocks and safety stocks, and it is current practice to compute them separately. Order cycle stocks depend on procurement lot sizes which are usually optimized with deterministic lot-sizing models. Safety stocks are required to protect against risk during the period of time which elapses between two supply orders and they depend on the variance of the demand process. Under these conditions, which are assumed to prevail in our context, roll sizes assortment and assignment decisions do not have a direct impact on the lot-sizing problem, but they have a significant impact on safety stock calculations because of risk pooling effects.

Consequently, one can rely on the literature for help on how to manage parent roll inventories, but the design problem of selecting the assortment of parent roll sizes to stock, and the assignment of parent rolls to finished products, has not been studied in any dept. This problem can be seen as a bill-of-material design problem. It is evident that the trim obtained when cutting a finished product of a given width depends on the width of the parent rolls from which it is sheeted. Keeping too many parent roll sizes, however, leads to high inventory holding costs. The basic trade-off between waste and inventory performance in a paper mill is discussed in Hameri (1995). The aim of this paper is to propose an optimization model and some efficient heuristics to find a parent roll size assortment, and assignments to finished products, minimizing expected trim and inventory holding costs. One of the solution approach proposed is a branch and price algorithm based on column generation and a fast pricing heuristic. The second solution method proposed is a greedy marginal cost heuristic. The two methods are tested on real data and also on randomly generated problem instances. The approach proposed was implemented by a large pulp and paper company.

2. Problem Definition

Our problem is to determine the best assortment of parent roll sizes to keep in stock and the parent roll size to use to sheet each type of finished product produced by a converting plant using one or several identical sheeters. In pulp and paper supply chains, the paper machines are usually the bottleneck. Paper companies own several converting plants but external converters can also be used to provide additional capacity if required. The assignment of finished products to converting plants is a tactical decision that precedes the assortment and assignment decisions considered here, so that the converting plant in our problem can be assumed to have enough sheeting capacity to cover all its finished products demand. The parent roll assortment and assignment problem considered is illustrated in Figure 2.

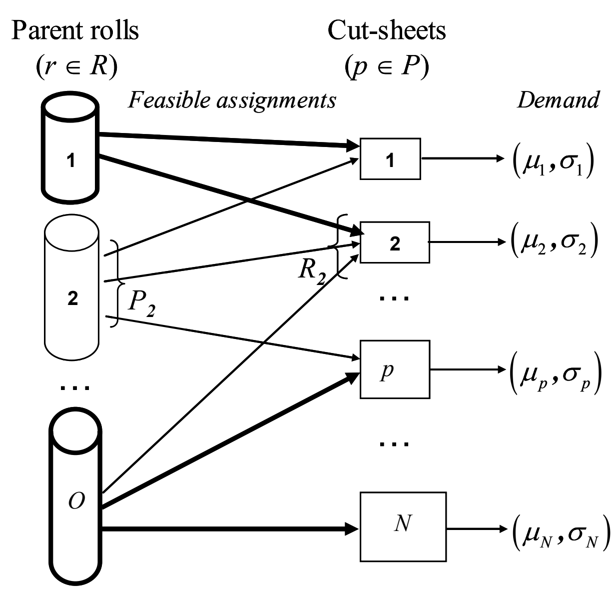


Figure 2: Potential Assortment and Assignments Graph

Let P be the set of finished products, R be the set of parent rolls which could possibly be stocked. Because of the technical characteristics of sheeters, storage shelf dimensions and mill operations policies, a given finished product $p \in P$ cannot be cut from any parent roll. Usually, only a few roll diameters (the diameter uniquely defines the unrolled paper length) can be considered and possible roll widths fall between a lower and an upper bound. Let R_p be the set of parent roll types which could be used to sheet product $p \in P$. Then, the set of the rolls which could be stocked is $R = \bigcup_{p \in P} R_p$. Note that the sets R_p, $p \in P$, implicitly define the technical constraints which must be respected. An assortment is a subset of R sufficient to sheet all the finished products demanded. An assortment, with feasible roll assignments to finished products, is illustrated with bold arcs and nodes in the graph of Figure 2. Any feasible assortment leads to expected inventory holding and trim loss costs and our objective is to find an assortment minimizing these costs. This warrants a closer examination of the inventory management and waste generation processes for our problem.

Several approaches can be used to compute safety stocks, depending on whether we want to impose a given service level or minimize inventory holding and shortage costs. Since shortage costs are difficult to estimate, it is current in practice to impose a predetermined service level. The service criterion most often used is to require that the probability that there will be no stock-outs during the relevant risk period is at least α. When this is done, under the additional assumption that the parent roll demand is Normally distributed, the safety stock for parent roll r can be computed with the expression $SS_r = k_\alpha \sqrt{Var_r}$. In this expression, k_α is the value of the standardized Normal variate which has probability α of not being exceeded, and Var_r is the variance of the demand for roll r during the risk period. In our case, since production on the paper machines is cyclic, the risk period is equal to the length of a production cycle of the paper grade considered plus the mill-to-converter transportation lead time. It is the expected holding cost of the safety stocks of the parent rolls in the assortment selected that must be taken into account in our optimization problem.

The different types of waste which may be generated when a parent roll is sheeted are illustrated in Figure 3. Note first that a fixed side trim is always lost on both sides of a parent roll when it is cut, independently of the cutting pattern used. In addition, if the roll width, net of the fixed trims, is not an exact multiple of the width of the required paper sheets, an additional trim loss is generated. Finally, when a cut-sheet order does not require the sheeting of a complete roll, an overrun is left. In what follows, we assume that the overrun left can always be used to satisfy other customer orders for the same type of paper so that it is not wasted. This implies that only trim loss has to be taken into account in our model. Note however that it is important to include the fixed trims in the waste calculation because they guarantee that the widest feasible roll is always selected. For example, suppose the cut-sheet width is 20 inches, and the fixed trim loss at the sheeter is 2 inches, then using parent rolls of (2*20 + 2) = 42 inches rather than rolls of (20 + 2) = 22 inches leads to less waste for a given order. If the fixed trim was not taken into account, these two options would be equivalent. By including fixed trim, one makes sure that sheeter capacity is not wasted.

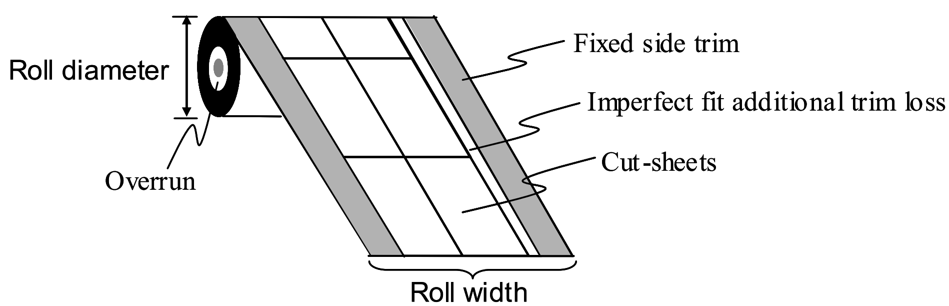


Figure 3: Cut-sheets and Trim for a Parent Roll

Given this, when the assignment (r, p) of a parent roll r to a finished product p is considered, it is easy to compute the unit trim loss cost for this assignment. It is customary in the industry to express paper demand and inventories in weight units. Without loss of generality, to simplify the presentation, we assume that weights are measured in pounds. Knowing the weight of a roll and the value of one pound of paper, the unit trim loss cost is obtained directly from the proportion of the roll width lost.

3. Model Formulation

Our objective in this work is to define product-roll assignments which minimize expected parent roll inventory holding costs and sheeter trim loss costs. Since the demand processes are stationary, these expected costs can be computed over a period of an arbitrary length. To simplify the presentation, the time period used to formulate the model is the parent rolls delivery lead time.

The following notations are needed to formulate the model:

R : Set of parent rolls ($r \in R$).

o : Total number of parent rolls i.e. $o = |R|$.

P : Set of finished products ($p \in P$).

n : Total number of products i.e. $n = |P|$.

P_r : Set of all the finished products that can be cut from roll r ($P_r \subset P$).

R_p : Set of all the parent rolls that can be used to produce finished product p ($R_p \subset R$).

v_r : Value of one pound of parent roll r paper

h_r : Inventory holding cost of one pound of parent roll r during a delivery lead time, i.e. $h_r = i\tau v_r$ where i is the annual inventory holding cost rate and τ is the delivery lead time in years.

f_{rp} : Number of pounds of roll r paper required to make one pound of finished product p, i.e. (parent roll r weight)/(weight of the sheets of product p cut from roll r).

c_{rp} : Trim loss cost associated to the production of one pound of product p with roll r, i.e. $c_{rp} = \left(f_{rp} - 1\right)v_r$.

μ_p : Mean product p demand, in pounds, during a delivery lead time.

σ_p : Standard deviation of the demand for product p during a delivery lead time.

Var_r : Variance of the demand for roll r during a delivery lead time.

k_α : (1-α) percentile of the standardized Normal variate.

x_{rp} : Binary decision variable equal to 1 if roll r is selected to make product p and to 0 otherwise.

Referring to Figure 2, it is seen that the assignment variables x_{rp} are associated to the arcs of the potential assortment and assignments graph, and that the problem to solve is a variant of the assignment problem. In order to formulate the objective function, expressions for the expected safety stock costs and trim loss costs during a delivery lead time must be derived. Since the demands for the finished products $p \in P$ are independent stationary stochastic processes with mean μ_p and variance σ_p^2, the mean and variance of the demand for parent roll *r*, in pounds, is given by the expressions:

$$Mn_r = \sum_{p \in P_r} f_{rp} \mu_p x_{rp} \qquad\qquad Var_r = \sum_{p \in P_r} f_{rp}^2 \sigma_p^2 x_{rp}$$

The safety stocks can then be calculated with the expression $SS_r = k_\alpha \sqrt{Var_r}$, as indicated before. Note that although this expression requires that the parent roll demand is Normally distributed, since several products are typically made from the same roll, the central limit theorem tells us that this condition will tend to be satisfied even if the demand for individual finished products is not Normally distributed. Note also that the safety stock obtained is expressed in pounds, and thus that it does not necessarily correspond to an integer number of parent rolls. This is normal because of the overrun associated to customer orders and because inventory control parameters can be expressed in pounds and not in rolls.

The calculation of the expected trim loss cost is straightforward. From the previous discussion, it is seen that the problem at hand can be formulated as follows:

$$Min \quad \sum_{r \in R} (h_r k_\alpha \sqrt{Var_r} + \sum_{p \in P_r} c_{rp} \mu_p x_{rp}) \qquad (P_1)$$

subject to

$$\sum_{r \in R_p} x_{rp} = 1, \ \ p \in P, \qquad (1)$$

$$Var_r = \sum_{p \in P_r} f_{rp}^2 \sigma_p^2 x_{rp}, \quad r \in R, \qquad (2)$$

$$x_{rp} \in \{0,1\},\ r \in R,\ p \in P_r \tag{3}$$

The objective function of P_1 sums expected inventory holding costs and trim loss costs for every parent roll. Constraints (1) stipulate that every finished product must be associated with one roll only. Constraints (2) compute the variance of the demand of selected rolls from the variance of the demand of their associated products. Constraints (2) can be used to eliminate Var_r from the model by substituting it in the objective function. The resulting model is a nonlinear integer program which can be difficult to solve for large problem instances.

4. Solution Methods

In this section, we introduce a branch-and-price approach based on column generation to solve large instances of the problem efficiently. To use this approach, problem (P_1) must first be reformulated as a set partitioning problem. A marginal cost heuristic which is much simpler to implement is also proposed.

4.1 Branch-and-price algorithm

A parent roll r can be used to produce any product $p \in P_r$ and the total cost associated with this roll is given by the expression:

$$C_r(x_r) = h_r k_\alpha \sqrt{\sum_{p \in P_r} f_{rp}^2 \sigma_p^2 x_{rp}} + \sum_{p \in P_r} c_{rp} \mu_p x_{rp} \tag{4}$$

where x_r is the roll r assignment vector $[x_{rp}, p \in P_r]$. Since x_r is a binary vector, 2^{n_r} $(n_r = |P_r|)$ assignment vector values can be considered for roll *r*. Let S_r be the index set of all possible assignments for roll *r* and $\{\hat{x}_r^s,\ s \in S_r\}$ be the possible assignments vector. The cost of assignment vector *s* for roll *r* is $C_{sr} = C_r(\hat{x}_r^s)$. From these observations, it is seen that model (P_1) can be reformulated as a set partitioning problem. Let $A_{sr} = [a_{1sr}, a_{2sr}, \ldots, a_{nsr}]$, be a N dimensional vector (column) with elements

$$a_{psr} = \begin{cases} \hat{x}^{s}_{rp} & \text{if } p \in P_r \\ 0 & \text{if } p \notin P_r \end{cases} \tag{5}$$

Also, let y_{sr} be a binary decision variable equal to 1 if assignment *s* is selected for roll *r*, i.e. if column A_{sr} is selected, and to 0 otherwise. Using this additional notation, model (P_1) can be reformulated as follow:

$$Min \sum_{r \in R} \sum_{s \in S_r} C_{sr} y_{sr} \tag{P$_2$}$$

subject to:

$$\sum_{r \in R} \sum_{s \in S_r} a_{psr} y_{sr} = 1, \; p \in P \tag{6}$$

$$\sum_{s \in S_r} y_{sr} \le 1, \; r \in R \tag{7}$$

$$y_{sr} \in \{0,1\} \quad r \in R, \; s \in S_r \tag{8}$$

The advantage of this set partitioning formulation is that it is an integer linear program (ILP), while the original model (P_1) was an integer nonlinear program. Although small instances of (P_2) can be solved as an ILP using commercial solvers such as Cplex, in practice the cardinality of the sets S_r is quite large and the ILP obtained is too big to be solved with these tools. Previous research on the solution of large generalized assignment problems has shown that the linear programming (LP) relaxation of the set partitioning formulation of the problem provides very tight bounds on the value of the optimal IP solution (Savelsbergh, 1997). However, since the number of variables (columns) in (P_2) increases exponentially with the number of finished products, the LP relaxation itself may be difficult to solve with standard LP software. An approach which has proven very efficient to solve large-scale LPs is column generation. This suggests that branch-and-price, which is a solution technique using column generation to solve LP relaxations in a branch-and-bound tree, should be an efficient approach to solve (P_2). It is this solution approach that is investigated in this section. For the application of branch and price methods and recent results the reader is refer to Savelsbergh (1997), Barnhart et al. (1998) and Hans (2001).

Before we start, to simplify the problem further, it is easy to show that:

Proposition 1: The optimal solution of the relaxed program

$$Min \sum_{r\in R}\sum_{s\in S_r} C_{sr} y_{sr} \text{ s.t. } \sum_{r\in R}\sum_{s\in S_r} a_{psr} y_{sr} = 1,\ p\in P;\ y_{sr}\in\{0,1\},\ r\in R,\ s\in S_r \qquad (P_3)$$

does not contain more than one column associated with any S_r.

Proposition 1 implies that it is sufficient to solve (P_3) to obtain the optimal roll assortment and assignments. In what follows, the linear programming relaxation of (P_3) is denoted by LP(P_3). In order to solve (P_3), we propose to start by solving LP(P_3) with a column generation procedure. A recent review of advances in column generation is found in Lübbecke and Desrosiers (2004). When LP(P_3) is solved by column generation, instead of calculating reduced costs explicitly for all the columns $A_{sr},\ r\in R,\ s\in S_r$ of the *master problem* LP(P_3) in the simplex method, we solve *restricted master problems* based on adequately selected small subsets $S'_r \subseteq S_r$ of columns for each roll *r* and, at each iteration of the procedure, we compute the reduced costs of the columns $A_{sr},\ r\in R,\ s\in S_r$ implicitly. The restricted master problem to solve at each iteration is:

$$Min \sum_{r\in R}\sum_{s\in S'_r} C_{sr} y_{sr} \text{ s.t. } \sum_{r\in R}\sum_{s\in S'_r} a_{psr} y_{sr} = 1,\ p\in P;\ y_{sr}\ge 0,\ r\in R,\ s\in S'_r \qquad \text{(RMP)}$$

Assuming that we have a feasible solution, let $y^*_{sr},\ r\in R,\ s\in S'_r$, and $u_p,\ p\in P$, be the primal and dual optimal solutions of (RMP), respectively. Since, for each variable y_{sr}, the cost coefficient C_{sr} can be computed with $C_r(x^s_r)$, where x^s_r is the assignment vector embedded in column A_{sr}, the *pricing problem* to solve to obtain the largest reduced cost is:

$$c^* = \underset{r\in R}{Max}\left\{c^{\circ}_r\right\};\ c^{\circ}_r = Max\left\{\sum_{p\in P_r} u_p x_{rp} - C_r(x_r)\,\middle|\, x_{rp}\in\{0,1\},\ p\in P_r\right\} \qquad \text{(PP)}$$

If $c^* \le 0$, no reduced cost is positive and $y^*_{sr},\ r\in R,\ s\in S'_r$, is the optimal solution of LP(P_3). Otherwise, we add to (RMP) the column A_{sr} embedding the assignment vector $\widehat{x}^s_r$ found by solving (PP), and proceed to the next iteration of the procedure by re-optimizing the revised restricted master problem. Let's denote the sub-problem to solve to get c°_r by (PP_r).

To simplify the presentation, (PP_r) can be rewritten as follows:

$$c_r^{\circ} = Max\ c_r(x_r) = \sum_{p \in P_r} b_{rp} x_{rp} - d_r \sqrt{\sum_{p \in P_r} g_{rp} x_{rp}}\, ;\ x_{rp} \in \{0,1\},\ p \in P_r \qquad (PP_r)$$

where $b_{rp} = u_p - c_{rp}\mu_p$, $d_r = h_r k_\alpha$ and $g_{rp} = f_{rp}^2 \sigma_p^2$.

PP_r is an integer non-linear program but of smaller dimensions. Note that the size of PP_r depends on the cardinality of the set of finished products P_r and it is difficult to solve when n_r is large. This motivated us to adopt a heuristic approach to solve the problem. Fortunately, for the column generation scheme to work, it is not necessary to select the column with the highest reduced cost; any column with a positive reduced cost will do. When the reduced cost used is not the highest, however, there is no guarantee that the optimal solution of LP(P_3) will be found.

Clearly, if the solution y_{sr}^*, $r \in R$, $s \in S_r'$ obtained by this column generation procedure is integral, it is optimal for (P_3). If it is not, however, we need to use a branch-and-bound procedure to obtain the optimal solution of (P_3). In each node of the tree, a reduced version of problem (RMP) is solved using the column generation procedure. Branching occurs when no column price out to enter the basis and the LP solution is not integral. Three crucial issues must be examined in more dept to implement this solution approach: i) the method used to construct an initial feasible restricted master problem, ii) the algorithm proposed to solve the pricing problems, and iii) the branching strategy. The remainder of this section is devoted to these issues.

4.1.1 Initial columns

In order to start the column generation procedure, a column A_{1r} must be provided for each parent roll $r \in R$. Also, the problem (RMP) defined by these initial columns must have a feasible solution to ensure that proper dual information is passed to the pricing problem. Appropriate initial columns can be obtained by first finding a feasible solution to problem

(P_1), and then by identifying the columns corresponding to this solution in (P_3). An easy way to find a feasible solution to (P_1) is to neglect the non linear part of its objective function, and to solve the resulting linear 0-1 programming problem. This new problem can be decomposed into the following n small problems:

$$Min \sum_{r\in R_p} c_{rp}\mu_p x_{rp} \text{ s.t. } \sum_{r\in R_p} x_{rp} = 1;\ x_{rp} \in \{0,1\},\ r \in R_p \qquad \text{P}(p)$$

Problem P(p) is trivial to solve by finding the parent roll $r \in R_p$ which has the smallest objective function value. Let $\hat{x}_r^1$ be the assignment vector for roll $r \in R$ required to construct initial column A_{1r} with (5). The required vectors are obtained with the following procedure.

Initialization heuristic: **(H$_1$)**

1. Find $r_p^* = \underset{r}{Min}\left\{c_{rp},\ r \in R_p\right\}$ for all $p \in P$.
2. Set $\hat{x}_{rp}^1 = \begin{cases} 1 & \text{if } r = r_p^* \\ 0 & \text{otherwise} \end{cases},\ p \in P,\ r \in R_p$.

4.1.2 Heuristic for the pricing sub-problem

Program (PP_r) is a convex integer maximization problem difficult to solve to optimality. In this sub-section, we develop an efficient heuristic which is able to find an optimal solution in most of the cases. To simplify the notation, we drop the index r in this section. Hence, the problem to solve is the following:

$$c^o = \underset{x}{Max}\ c(x) = \sum_{p\in P} b_p x_p - d\sqrt{\sum_{p\in P} g_p x_p},\ x_p \in \{0,1\}, p \in P \qquad \text{(PP')}$$

where d and g_p, $p \in P$, are positive coefficients, but where b_p, $p \in P$, can take any value. Consequently, depending on the value of the b_p's, c^o could be non-positive, in which case no new interesting column can be obtained by solving (PP'). Also, for c^o to be positive, at least one b_p must be positive. For a given x, the function $c(x)$ has the form $c(g,b) = -d\sqrt{g} + b$, where $g = \Sigma_{p\in P} g_p x_p$ and $b = \Sigma_{p\in P} b_p x_p$, i.e. where g and b can take a finite set of values depending on x. In order to explain our pricing heuristic some properties of this function must first be derived.

Proposition 2: Let $c(g,b) = -d\sqrt{g} + b$, where $g \geq 0$ and d is a positive number. The following statements are true:

1. If $c(g^*,b^*) \geq 0$, then $c(g+g^*,\, b+b^*) \geq c(g,b)$.
2. If $c(g^*,b^*) < 0$ and $2d\sqrt{g} \geq (d^2 g^*/b^*) - b^*$, $b^* \geq 0$, then $c(g+g^*,\, b+b^*) \geq c(g,b)$.
3. If $c(g^*,b^*) < 0,\ b^* \leq 0$ then $c(g+g^*,\, b+b^*) \leq c(g,b)$.
4. If $c(g_1,b_1) = c(g_2,b_2)$, $b_1 \leq b_2$ then $c(g+g_1, b+b_1) \leq c(g+g_2, b+b_2)$.

Proof: See the appendix.

Proposition 2 implies that if one wanted to construct a solution vector x maximizing $c(x)$ by sequentially setting the x_p's equal to 1 until the value of $c(x)$ cannot be increased anymore, one should start with the products p having the largest $F_p = b_p - d\sqrt{g_p}$, $p \in P_r$. Let $B(j)$, $j = 1,...,n$, be the list, of length $n = |P|$, of product indexes sorted, first, in decreasing order of $F_p = b_p - d\sqrt{g_p}$, $p \in P$ (i.e. such that $F_{B(j)} \geq F_{B(j+1)}$), and second, in case of ties, in decreasing order of b_p's (i.e. such that $b_{B(j)} \geq b_{B(j+1)}$). Also, for the first k terms of that list, define $b(k) = \sum_{j=1}^{k} b_{B(j)}$ and $g(k) = \sum_{j=1}^{k} g_{B(j)}$. Given this, the following proposition is true.

Proposition 3: If $b(k) + b_{B(k+1)} - d\sqrt{g(k) + g_{B(k+1)}} \leq b(k) - d\sqrt{g(k)}$, then $b(k) + b_{B(k+2)} - d\sqrt{g(k) + g_{B(k+2)}} \leq b(k) - d\sqrt{g(k)}$ when $g_{B(k+2)} \leq g_{B(k+1)}$.

Proof: See the appendix.

Proposition 2 and Proposition 3 help us to develop an algorithm to generate a column by sequentially including products which at least guarantees to improve the reduced cost. The algorithm defines a subset, $K \subseteq \{1,..., j,..., n\}$, of the product index list, which identifies the elements $B(j)$ of the assignment vector x to set to 1. At the termination, if K is empty, then there is no column which can price out the columns in the current basic solution of (RPM).

Pricing heuristic: **(H_2)**

1. Set $K = \varnothing$, $i = 1$ and $c^{o} = -\infty$.
2. Construct the list of indexes $B(j)$, $j = 1,...,n$, by sorting the product indexes in decreasing order of $F_p = b_p - d\sqrt{g_p}$ and b_p.
3. While $i \notin K$ and $\sum_{k \in K} b_{B(k)} + b_{B(i)} - d\sqrt{\sum_{k \in K} g_{B(k)} + g_{B(i)}} \geq c^{o}$, do

   $K = K \cup \{i\}$.

   $c^{o} = \sum_{k \in K} b_{B(k)} - d\sqrt{\sum_{k \in K} g_{B(k)}}$.

   $i = i + 1$.

   EndWhile.
4. For $j = i+1, \ldots, n$, do:

   If $g_{B(j)} > g_{B(i)}$ and $j \notin K$ and $\sum_{k \in K} b_{B(k)} + b_{B(j)} - d\sqrt{\sum_{k \in K} g_{B(k)} + g_{B(j)}} \geq c^{o}$, then

   $K = K \cup \{j\}$.

   $c^{o} = \sum_{k \in K} b_{B(k)} - d\sqrt{\sum_{k \in K} g_{B(k)}}$.

   Go to 3.

   EndFor.
6. Set $x_{B(j)} = \begin{cases} 1 & j \in K \\ 0 & j \notin K \end{cases}$, $j = 1,...,n$.

We tested (H_2) on several randomly generated problems. To find the optimal solution to the problem instances we enumerated all the possible solutions. Since complete enumeration is exponential (2^n possible solutions) we restricted ourselves to n=10 products problems. We solved 1,000 randomly generated instances and found only one instance where (H_2) did not reach the optimal solution. Although the same performance can not be guaranteed for large instances, it appears that using our pricing heuristics leads to the optimal solution in most cases.

4.1.3 Branching strategy

As indicated previously, the optimal solution of problem LP(P_3) may contain several fractional variables, in which case we need to use a branch and price procedure to find the optimal solution of (P_3). Furthermore, the rounding of the solution obtained for a given node of the B&P tree may not give a feasible solution. In this case it is mandatory to generate new columns to get a feasible solution. If any decision variable is fractional, then at least two finished products are allocated to at least two different parent rolls. Two branching strategies can be adopted in this context. The first one selects a product assigned to two or more parent rolls and assigns it to one of these parent roll, for the left branch, and avoids it for the right branch. The second strategy assigns a subset of products to the same parent roll, for the left branch, and avoids this subset of products for the right branch. Every time a pricing problem is solved for a given node of the B&P tree, one has to check if it violates these product assignment restrictions. If they do, revised columns are generated by altering the product assignments. Since the later branching strategy assigns several products to a parent roll, it turns outs that we get a feasible solution very quickly and thus an integer bound. The disadvantage of this strategy, however, is that when the columns generated by the pricing problems are not valid for the right branch, we need to generate 2^{nd}, 3^{rd}… best combinations (in terms of reduced price) to find a good valid column. In our implementation we use a mixed strategy: we select the column with the highest fractional value and we fix at most two products in this column for the left branch, and make sure that they are not present together in the right branch. Finally, we use depth first search in our implementation.

4.2 Marginal cost heuristic

Although the column generation approach proposed gives very good results, as will be shown in the next section, it is difficult to implement. This motivated us to develop a simple and fast heuristic which could be easily implemented, and to investigate the quality of the solution it provides. The solution method elaborated is a greedy heuristics which starts with a feasible solution and improves it gradually. At each iteration, new solutions in the neighborhoods of the current best solution are constructed by switching assignment arcs, and their cost is computed to select the best one. To get an initial feasible solution we use the initialization heuristic (H_1) presented in section 4.1.1. The algorithm stops when no improved solution is found.

Let $\hat{x}_r^i,\ r \in R$, be the feasible solution obtained at the i^{th} iteration of the algorithm and let $C_{ir} = \sum_{r \in R} C_r(\hat{x}_r^i)$ be the cost of this solution. The detailed algorithm follows.

Marginal cost heuristic: **(H_3)**

1. Set $i = 1$, construct $\hat{x}_r^1,\ r \in R$, with heuristic H_1, compute C_{1r} and set $C_{\min} = C_{1r}$.
2. Define solution i arc set $X^i = \left\{(r,p) \mid \hat{x}_{rp}^i = 1, r \in R, p \in P_r\right\}$ and set $j = 0$.
3. For all $(r,p) \in X^i$, do:

   For all $r' \in R \setminus r$ do:

   Set $j = j + 1$ and define $\hat{x}_r^{i(j)} = \hat{x}_r^i,\ r \in R$.

   Replace arc (r,p) by arc (r',p), i.e. set $\hat{x}_{rp}^{i(j)} = 0,\ \hat{x}_{r'p}^{i(j)} = 1$.

   If $\sum_{r \in R} C_r(\hat{x}_r^{i(j)}) < C_{\min}$ then $j_{\min} = j$ and $C_{\min} = \sum_{r \in R} C_r(\hat{x}_r^{i(j)})$.

   Enddo.

   Enddo.
4. If $C_{\min} < C_{ir}$ then set $i = i + 1,\ C_{ir} = C_{\min},\ \hat{x}_r^i = \hat{x}_r^{i-1(j_{\min})},\ r \in R$, and go to step 2.
   Else, stop.

Note that heuristic (H_3) can be used instead of (H_1) to initiate the column generation procedure. Also, the solution obtained provides an upper-bound on the value of the optimal solution which can be used to improve the branching process of the B&P algorithm.

5. Experimental Testing

In this section we present two sets of numerical experiments performed to test the performances of our solution methods. The Branch and Price algorithm was implemented with VC++.net and CPLEX 9.0 and the experiments were run on a 1.8 Ghz computer with 512 MB of RAM. In the first set of experiments, we solved real problems using data from one of the largest fine paper mill in North-America, and we calculated overall gains over the managerial rule used by the company decision-makers. We used one year sales data for different finished products, and found the optimal parent roll assortment with our B&P algorithm, for each paper grade produced by the mill. The results obtained are summarized in Table 1. The use of our approach reduces the number of rolls in the mill assortments from 75 to 53. This yields a 29.34 % reduction in inventory holding costs and a 1.72% reduction in trim loss costs. It is interesting to note that the mill managers had traditionally concentrated their efforts on trim loss reduction, and that they were not considering inventory holding costs in their roll assortment decisions. Our results show that the use of our algorithm gives comparable results in terms of trim loss but with considerable savings in inventory holding costs.

Table 1: Roll Assortment Optimization Problems for Different Paper Grades

Paper grade	Problem size ($R \times P$)	Number of rolls in optimal assortment	Processing time in seconds
1	9×4	2	3
2	10×4	4	2
3	12×11	6	23
4	55×28	7	12
5	32×18	9	24
6	61×28	11	24
7	53×36	14	15

In the second set of experiments, we solved different instances of the problem, using randomly generated data, to compare the performances of the marginal cost heuristic (H_3) and of the B&P algorithm, when (H_3) is used to initialize the procedure and to calculate an upper-bound. Twenty two different problem sizes were considered and, for each of them, 25 problem instances were solved using the two solution methods. The percentage cost difference and computing time difference between the solutions given by the two methods were calculated. The mean and standard deviation of the percentage cost difference, and the mean difference in computing times, are presented in Table 2. The gains presented are calculated as follows: Percentage gain = $100 * (\text{B\&P time} - \text{H}_3\text{ time}) / \text{B\&P time}$). The last column of Table 3 provides the averaged number of nodes which had to be explored in the B&P algorithm to get the solution. For the 25 instances of the largest problems solved (60 × 40), the computation times of the marginal cost heuristic were in the [1.56, 2.57] seconds range, with an average of 1.93 seconds. For the B&P algorithm, they were in the [7.0, 140.9] seconds range with an average of 40.28 seconds. Note, that these problems were also solved by initiating the B&P algorithm with heuristic (H_1) instead of (H_3). It was found that using (H_3) gives computing times which are 56% lower, on the average.

Table 2: Solution Methods Comparison

	Problem size ($R \times P$)	Mean percentage difference (MPD) (H_3 – B&P)/B&P	Standard deviation of MPD	Processing time gain of heuristic over B&P (%)	Nodes solved (Average in nearest integer)
1	5 × 10	0.132	0.50	92.20	3
2	5 × 20	0.23	0.54	95.74	3
3	6 × 10	0.080	0.27	84.31	3
4	6 × 20	0.221	0.48	92.52	4
5	7 × 10	0.14	0.53	84.15	3
6	7 × 20	0.43	0.79	92.07	4
7	8 × 10	0.20	0.74	89.25	3
8	8 × 20	0.557	0.85	95.02	4
9	9 × 10	0.33	0.91	93.59	3
10	9 × 20	0.94	1.70	93.59	4
11	10 × 10	0.46	1.07	89.72	3
12	10 × 20	1.00	1.15	93.49	5
13	20 × 15	7.31	5.46	90.17	4
14	20 × 40	7.36	5.15	94.19	6
15	30 × 15	10.0	8.14	83.21	4
16	30 × 40	12.19	7.42	91.28	10
17	40 × 15	10.62	9.52	79.04	4
18	40 × 40	14.83	10.83	90.60	9
19	50 × 15	11.62	11.38	78.53	4
20	50 × 40	12.74	9.52	85.97	7
21	60 × 15	8.11	8.21	76.66	4
22	60 × 40	14.24	10.66	87.74	8

Since, as discussed in section 4.1.2, pricing heuristic (H_2) almost always gives an optimal solution to the pricing sub-problem, in most cases, the solution given by the B&P algorithm is optimal. On the other end, from the experiment results, we can observe that as the problem size increases the mean cost difference between the two solution methods increases. In other words, the marginal cost heuristic may not give consistent performances for very big problems. On the other hand, using the marginal cost heuristic in the B&P procedure significantly improves its efficiency for big instances of the problem. For the largest problem solved, 1390 nodes are explored when using (H_1) in the B&P algorithm, but using the marginal cost heuristic solution as an upper bound brings this number down to 67 nodes. In most of the small instances, 3 nodes only, including the root node, must be explored to find the solution.

6. Concluding Remarks

This paper proposes an optimization model and some efficient heuristics to find a parent roll assortment and assignments minimizing expected trim and inventory holding costs in a fine paper mill. The binary non-linear programming model proposed is an essential element of a sheet-to-order supply chain strategy, which is a very attractive alternative to the current order penetration point strategies used by paper companies. The branch and price solution approach proposed to solve a set-partitioning reformulation of the problem almost always gives the optimal roll assortment and assignments. The marginal cost heuristic proposed is much simpler to implement than the B&P algorithm, and much faster, but it gives solutions which are slightly more expensive, on average, for realistic size problems. However, given its simplicity, for the problem size encountered in the mills we have studied, it is a very attractive practical alternative. In fact, it is this algorithm that the company we were working with decided to implement.

References

[1] Barnhart, C., E, Johnson, G. Nemhauser, M. Savelsbergh and P. Vance (1998). Branch-and-price: Column generation for solving huge integer programs, *Operations Research* 46(3), 316-329.

[2] Correia, M. H., J. F. Oliveira and J. S. Ferreira (2004). Reel and sheet cutting at a paper mill, *Computer & Operations Research* 31, 1223-1243.

[3] Dyckhoff, H. (1990). A typology of cutting and packing problems, *European Journal of Operational Research* 44(2), 145-159.

[4] Goulimis, C. (1990). Optimal solutions to the cutting stock problem. *European Journal of Operational Research* 44(2), 197–208.

[5] Hameri, A.-P. (1995). Efficient reel inventory control - a trade-off between waste and inventory performance, *Paper and Timber* 77(8), 479-482.

[6] Hameri, A.-P. and J. Nikkola (2001). Order penetration point in paper supply chains, *Paper and Timber* 83(4), 299-302.

[7] Hans, E. (2001). *Resource loading by Branch-and-Price techniques*. Ph. D. thesis, Beta Research School for Operations Management and Logistics.

[8] Keskinocak, P., F. Wu, R. Goodwin, S. Murthy, R. Akkiraju, S. Kumaran and A. Derebail (2002). Scheduling solutions for the paper industry. *Operations Research* 50(2), 249-259.

[9] Kginaricha, E., R. Rubio, M.. Taksar and L. Wein (1998). A Dynamic Stochastic Stock-cutting Problem, *Operations Research*, 46(5), 690-701.

[10] Lehtonen, J.-M. and J. Holmström (1998). Is just-in-time applicable in paper industry logistics?, *Supply Chain Management* 3(1), 21-32.

[11] Lodi, A., S. Martello and M. Monaci (2002). Two-dimensional packing problems: A Survey, *European Journal of Operational Research* (141), 241-252.

[12] Lübbecke, M. and J. Desrosiers (2004). Selected Topics in Column Generation, *Les Cahiers du GERAD* G–2002–64.

[13] Savelsbergh, M. (1997). A Branch-and-Price Algorithm for the Generalized Assignment Problem, Operations Research 45(6), 831-841.

[14] Sweeney, E. and R. W. Haessler (1990). One-dimensional cutting stock decisions for rolls with multiple quality grades, *European Journal of Operational Research* 44, 224-231.

[15] Silver, E., D. Pyke and R. Peterson (2003). *Inventory Management and Production Planning and Scheduling*, Wiley.

[16] Rizk, N., A. Martel and S. D'Amours (2005). Multi-Item Dynamic Production-Distribution Planning in Process Industries with Divergent Finishing Stages, fortcomming in *Computers and Operations Research*.

[17] Westerlund, T., I. Harjunkoski and J. Isaksson (1998). Solving a production optimization problem in a paper-converting mill with MILP, *Computers and Chemical Engineering*, 22(4/5), 563-570.

Appendix: Proof of Propositions

Proof of Proposition 2:

1. Assume $b^* - d\sqrt{g^*} \geq 0$ and $c(g+g^*, b+b^*) \leq c(g,b)$. Then,

$$d\sqrt{g+g^*} - d\sqrt{g} \geq b^* \tag{p1}$$

   Since all g, g^* and d are non-negative, $d\sqrt{g+g^*} - d\sqrt{g} = \varpi \geq 0$, and $d\sqrt{g+g^*} = \varpi + d\sqrt{g}$. The square of this expression gives the following:

$$d^2(g+g^*) = \varpi^2 + d^2 g + 2\varpi d\sqrt{g}\ , \text{ or } d^2 g^* = \varpi^2 + 2\varpi d\sqrt{g}\ .$$

   Consequently, $d\sqrt{g^*} \geq \varpi$ and, from (p1), $d\sqrt{g^*} \geq d\sqrt{g+g^*} - d\sqrt{g} \geq b^*$. This implies that $d\sqrt{g^*} \geq b^*$, which contradicts our original assumption.

2. Assume $2d\sqrt{g} \geq (d^2 g^*/b^*) - b^*$ is true. Since $b^* \geq 0$, rearranging terms, we obtain:

$$d^2 g^* \leq (b^*)^2 + 2db^*\sqrt{g}\ .$$

   Dividing by d^2, adding g on both sides, and taking the square root, we get:

$$\sqrt{g+g^*} \leq \sqrt{g} + \frac{b^*}{d}\ ,$$

   which implies that $-d\sqrt{g+g^*} + b^* \geq -d\sqrt{g}$. Adding b on both sides we have:

$$c(g+g^*, b+b^*) \geq c(g,b)\ .$$

3. Let us assume $c(g+g^*, b+b^*) \leq c(g,b)$ is true. Then, $d\sqrt{g+g^*} - d\sqrt{g} \geq b^*$. Since both g and g^* are positive, this implies that $d\sqrt{g} + d\sqrt{g^*} - d\sqrt{g} > b^*$, an thus that $c(g^*, b^*) < 0$.

4. When $c(g_1 + b_1) = c(g_2 + b_2)$ and $b_2 \geq b_1$, we have $-d\sqrt{g_1} + b_1 = -d\sqrt{g_2} + b_2$ and thus:

$$b_2 - b_1 = d(\sqrt{g_2} - \sqrt{g_1}) \geq 0 \tag{p2}$$

which implies that $g_2 \geq g_1$. Now assume that $c(g+g_1, b+b_1) - c(g+g_2, b+b_2) = \delta$, i.e. that:

$$-d\sqrt{g+g_1} + d\sqrt{g+g_2} + b_1 - b_2 = \delta \tag{p3}$$

Since $g_2 \geq g_1$, we can define $\sigma \geq 0$ such that $g_2 = g_1 + \sigma$. Now for a given g_1 and σ (p3) can be expressed as a function of g, i.e. we can define the function:

$$f(g) = -d\sqrt{g+g_1} + d\sqrt{g+g_1+\sigma} + d\sqrt{g_1} - d\sqrt{g_1+\sigma}$$

Differentiating $f(g)$ w. r. t. g, we get:

$$\frac{df}{dg} = \frac{\sqrt{g+g_1} - \sqrt{g+g_1+\sigma}}{2\sqrt{(g+g_1)(g+g_1+\sigma)}}d \leq 0$$

The above derivative shows that, for a positive g, function $f(g)$ is strictly decreasing. For a given g_1 and σ, the maximum value of $f(g)$ is zero when $g = 0$. This implies that $\delta \leq 0$ for a non-negative g.

This completes the proof. ■

Proof of Proposition 3:

Assume that $g_{B(k+2)} \leq g_{B(k+1)}$ and that

$$b(k) + b_{B(k+1)} - d\sqrt{g(k) + g_{(k+1)}} < b(k) - d\sqrt{g(k)} \tag{p4}$$

$$b(k) + b_{B(k+2)} - d\sqrt{g(k) + g_{B(k+2)}} \geq b(k) - d\sqrt{g(k)} \tag{p5}$$

Since the list B is sorted in decreasing order of F_p's, we have:

$$b_{B(k+1)} - d\sqrt{g_{B(k+1)}} \geq b_{B(k+2)} - d\sqrt{g_{B(k+2)}} \tag{p6}$$

Subtracting $d\sqrt{g_{B(k+1)}}$ and $d\sqrt{g_{B(k+2)}}$ on both sides of (p4) and p(5), respectively, we have:

$$b_{B(k+1)} - d\sqrt{g_{B(k+1)}} < d\sqrt{g(k) + g_{B(k+1)}} - d\sqrt{g(k)} - d\sqrt{g_{B(k+1)}} \qquad \text{(p7)}$$

$$b_{B(k+2)} - d\sqrt{g_{B(k+2)}} \geq d\sqrt{g(k) + g_{B(k+2)}} - d\sqrt{g(k)} - d\sqrt{g_{B(k+2)}} \qquad \text{(p8)}$$

From (p6), (p7) and (p8) we deduce

$$\sqrt{g(k) + g_{B(k+1)}} - \sqrt{g(k) + g_{B(k+2)}} - \sqrt{g_{B(k+1)}} + \sqrt{g_{B(k+2)}} > 0 \qquad \text{(p9)}$$

Define $f(g) = \sqrt{g + g_{B(k+1)}} - \sqrt{g + g_{B(k+2)}} - \sqrt{g_{B(k+1)}} + \sqrt{g_{B(k+2)}}$. The first derivative of this function w. r. t. g is:

$$\frac{df}{dg} = \frac{\sqrt{g + g_{B(k+2)}} - \sqrt{g + g_{B(k+1)}}}{2\sqrt{(g + g_{B(k+1)})(g + g_{B(k+2)})}} \qquad \text{(p10)}$$

Relation (p10) shows that $f(g)$ is a decreasing function when $g_{B(k+2)} \leq g_{B(k+1)}$. Furthermore, g is always non-negative and thus the maximum of $f(g)$ will be zero at $g = 0$, which contradicts relation (p9). This shows that inequality (p5) is not true, which completes the proof. ■