Interfaces
Vol. 41, No. 5, September–October 2011, pp. 455–465
ISSN 0092-2102 | EISSN 1526-551X | 11 | 4105 | 0455

http://dx.doi.org/10.1287/inte.1110.0590

Kimberly-Clark Latin America Builds an Optimization-Based System for Machine Scheduling

Nazrul Shaikh
Department of Industrial Engineering, College of Engineering, University of Miami,
Coral Gables, Florida 33146, n.shaikh@miami.edu

Vittal Prabhu
Marcus Department of Industrial and Manufacturing Engineering, Pennsylvania State University,
University Park, Pennsylvania 16802, prabhu@engr.psu.edu

Danilo Abril, David Sánchez
Kimberly-Clark, Latin American Operations, Bogotá, Colombia
{danilo.abril@kcc.com, david.r.sanchez.c@gmail.com}

Jorge Arias
Kimberly-Clark, Latin American Operations, C1001ABR Buenos Aires, Argentina, jorge.arias@kcc.com

Esteban Rodríguez
Kimberly-Clark, Latin American Operations, Heredia, Costa Rica, esteban.rodriguez@kcc.com

Germán Riaño
Kimberly-Clark, Latin American Operations, Bogotá, Colombia, german.riano@kcc.com

During a single planning period, Kimberly-Clark Latin America manufactures dozens of stock-keeping units (SKUs) in varying quantities using a few machines. The same SKU can be manufactured on multiple machines, some of which are more efficient than others. In addition, the setup time for an SKU is sequence dependent, and its demand is stochastic between planning periods. The stochastic demand necessitates changing production plans each planning period; given the large number of SKUs and small number of machines, this leads to inefficiencies. This paper describes the formulation and corresponding solution approach of an integrated inventory, production-planning, and detailed scheduling model to address the inefficiencies in lot sizing, production scheduling, and inventory management. The paper's key contribution is the solution approach, which solves the resultant industry-size NP-hard problem in minutes. The solution quality and its implementation have been tested extensively, and the model has been successfully deployed in five countries. A reduction in finished product inventories of up to 45 percent, an increase in yield and uptime of 2 percent, and improvements in service levels of 2.4 percent are directly attributable to the model and the solution approach highlighted in the paper.

Key words: production scheduling: sequencing; programming: integer; applications: production scheduling; lot sizing.

Kimberly-Clark (K-C) is a multinational corporation with operations in 35 countries and global brands that are sold in more than 150 countries. K-C's Latin American Operations (K-C LAO) division, which encompasses 22 countries south of Mexico, focuses on bringing various feminine, infant, adult-care, and other personal hygiene products to market. To cater to the diverse needs of the region's population, K-C LAO maintains an expansive portfolio of over 1,000 stock-keeping units (SKUs). Most of these SKUs satisfy niche demands; because competitors' products are usually available as substitutes, K-C LAO must keep a large product portfolio and a high fill rate. Furthermore, the competitive structure of the market causes low margins on products.

In 2007, K-C LAO had an expansive portfolio and was successful at maintaining high product availability; however, its inventory levels were high. The root cause of the inefficiency was found to be the stochasticity of demand at the SKU level. The stochastic demand necessitates that production plans be changed frequently, and that the production runs (batch sizes) change. A mathematical model based on production planning and detailed scheduling seemed to be an attractive approach to address the inefficiencies.

The mathematical model would take as an input the expected demand for multiple SKUs over a finite time horizon, the current inventory, an end-of-month inventory policy, production and inventory holding costs, machine capabilities, and changeover times. As output, the model would recommend the quantity of each SKU to manufacture, the machine to use, and the manufacturing sequence to follow on each machine during the entire planning horizon, while ensuring that overall profits are maximized. Such mathematical models are variants of the economic lot-scheduling problem (ELSP) that various researchers have studied extensively. However, ELSP and its variants are an NP-hard class of computationally intractable problems; therefore, they cannot be solved to optimality for large industrial-scale problems (Hsu 1983).

We formulated the integrated production planning, inventory management, and detailed scheduling problem as a mixed-integer programming (MIP) problem. The key contribution of the paper lies in the solution strategy we developed for solving the large-scale MIP; it solves the resultant industry-size NP-hard problem in minutes.

The solution strategy depends on three algorithms: (1) the assignment algorithm, which filters the production capabilities based on a linear programming (LP)-based assignment model; (2) the arc generation algorithm, which reduces the size of the solution space for the ELSP problem; and (3) the successive machine inclusion algorithm, which allows the inclusion of machines one at a time, practically reducing an M-machine problem to M single-machine problems, which are easier to solve. Together, these three algorithms ensure a scalable solution approach that is solvable within minutes.

We organized this paper as follows. We start with a brief *Literature Review* section covering the relevant work in ELSP. In the *Optimization Model* section, we present the model in some detail; Appendix A includes the exact formulation and its description. In the *Solution Strategy* section, we provide an overview of our heuristic for solving the model presented in the previous section; in Appendix B we give the details. In the *Results* section, we compare some key measures of performance of the best solution obtained to those obtained by experienced production planners, and the impact of the model usage. Finally, we present a section on *Conclusions and Ongoing Improvements*.

Literature Review

The classic ELSP involves developing a schedule for the production of several products on a single facility so that demand levels are met without back orders, and long-run average inventory-carrying and setup costs are minimized. It also assumes deterministic demand and production rates, and sequence-independent setup times and costs. The ELSP has been studied for more than 40 years; Elmaghraby (1978), Potts and Wassenhove (1992), Jans and Degraeve (2007, 2008) provide details related to various solution approaches. The key issue with the ELSPs is that they are computational complex problems; even the classic ELSP with restrictive assumptions has been shown to be NP-hard (Hsu 1983). Variants of the ELSP problem are at least NP-hard; some, such as those applied to the semiconductor manufacturing units, are NP-complete (Denton et al. 2006).

Given the complexity, any opportunities to exploit the structure of the problem and the solution space are generally used to obtain good-quality solutions. The ELSP is usually formulated as an MIP; heuristics are then developed to solve it. Belvaux and Wolsey (2000) describe a branch-and-cut approach based on combining several types of cuts to systematically reduce the size of the solution space. Belvaux and Wolsey (2001) describe methods for reformulating lot-sizing problems to achieve substantially improved computational times. Degraeve and Jans (2007) discuss a new Dantzig-Wolfe reformulation and branch-and-price algorithm to find a solution. Other approaches that have been widely studied include: the use of Lagrangian relaxation combined with branch-and-bound (Afentakis and Gavish 1986, Diaby et al. 1992); use of one-pass greedy-type approaches (Dixon and Silver 1981); metaheuristics such as tabu search (Simpson and Erengüç 1998); and column-generation approaches (Vanderbeck 1999).

More recently, the focus has moved from solving small academic exercise problems to larger industrial-scale problems by MIP reformulations using commercial MIP solvers, as discussed by Nemhauser et al. (1994), Belvaux and Wolsey (2000, 2001), and Cordier et al. (1999). Denton et al. (2006) discuss the solution method for an IBM case for a semiconductor manufacturing unit using an "MIP + heuristic" approach, in which the heuristic exploits the structure of the problem to solve the industry-scale problem; such problems also combine problem decomposition with reformulation.

The key concept behind decomposing the ELSP is that, if a heuristic can solve the variant of the ELSP for a single-machine case, it can be extended for the more general multimachine case under some assumptions. For example, the multimachine case in which parallel processors are available for production has been successfully solved. Most of the solution approaches in the existing literature make an initial product allocation and then separately solve the problem for each machine. Carreno (1990) considered the ELSP for the case of identical parallel machines and proved that if machines are identical the optimal allocation does not allow product sharing between machines. Bollapragada and Rao (1999) considered the non-identical parallel-machines version, and developed a concave minimization model that confines solutions to rotation schedules. Pesenti and Ukovich (2003) tackle the nonrelated parallel machines in which some products cannot be produced simultaneously; they use a two-step methodology in which products are assigned using a linear program, and sequence and lot sizes are proposed by a heuristic based on graph theory.

Two lessons from prior research on ELSP are relevant to the methodology developed at K-C LAO.

1. An effective solution to a lot-sizing problem depends on the development of a tight formulation for the specific problem (Belvaux and Wolsey 2001).

2. Most ELSP problems can be decomposed into a set of single-item problems with linking constraints (Wolsey 2002).

Our work is in line with the work by Belvaux and Wolsey (2000, 2001), Wolsey (2002), and Denton et al. (2006).

Optimization Model

Most of K-C's products in K-C LAO must compete for a limited number of machines, which must be set up whenever the SKU that is to be manufactured in the machine changes. These changeovers could require a lengthy process, because many parts of the machine must be dismounted and replaced by new parts. Furthermore, after starting the machine, a long tune-up time is required, wasting much raw material. Therefore, K-C LAO has a strong motivation to choose large batch sizes. However, although the long production batches reduce the number of setups, they increase inventory and inventory holding costs, and the manufacturing system becomes less efficient at satisfying the demands for specific product mixes. The optimization model aims to find the balance between batch sizes and changeovers that maximizes the firm's profits over an extended planning horizon.

We developed an MIP that takes as input the monthly demand (which is split in four weeks based on historical data) for each SKU, current inventory levels at the SKU level, target inventory levels for every SKU, production parameters (e.g., machine speeds, setup costs, and times for an SKU on a machine), SKU processing time by machines, production and inventory holding costs, and estimated backorder costs. Using these data, the model produces sequence and lot-size recommendations. The objective function considers the selling price, manufacturing costs, inventory holding costs, and backorder costs. Appendix A shows the details of the mathematical formulation.

Solution Strategy

We were able to solve the problem presented in Appendix A for many instances. However, larger plants with more SKUs generated instances that required some ingenuity to find a solution in less than an hour. Our solution strategy is built on applying the following three heuristic algorithms sequentially to speed up the solution. We explain the basic concepts behind the results and present their impact below. Appendix B provides the mathematical details.

1. The assignment algorithm reduces the universe of possible assignments of parts to machines, thus reducing the number of binary variables. The assignment algorithm uses information about the machine

capabilities (e.g., which SKUs the machine can manufacture), time that the machine requires to make each SKU, and corresponding cost, inventory state, demand, and costs in an LP formulation to assign lot sizes of SKUs to be manufactured on a given machine. In essence, it solves the optimization problem in Appendix A, albeit without including the constraints that introduce the sequence-dependent setups into the model. The resulting LP can be solved in a short time, and it provides us with a new allocation of SKUs to machines; we use this allocation to redefine the capabilities, by including for each machine only those products with a positive assignment quantity in the solution of the assignment model.

2. The arc generation algorithm reduces the universe of possible SKUs transitions by keeping only those that are close to the ideal sequence. This ideal sequence is obtained by solving a traveling salesman problem (TSP) for each machine, where each city is an SKU and the travel times are the costs incurred when making a setup from one SKU to another. Using this algorithm has an additional advantage: the resulting sequences do not vary significantly from one month to the next. To test the impact of using a subset of arcs rather than the whole set, we ran instances for four machines. We found that, on average, using 50 percent of the arcs, the model attains a solution value that is 99.5 percent of the optimal value, but uses only 82.4 percent of the original time. Figure 1 shows the evolution of the objective function and solution time as a function of the arc generation percentage.

3. The successive machine inclusion algorithm speeds up the problem's solution by including one machine at a time. Although no machine is capable of producing every product, some products can be made in multiple machines. This implies that to get the optimal solution, all the machines must be scheduled at the same time. However, because of the combinatorial nature of the problem, it is faster to solve a problem for each machine independently than to solve a single problem for all machines at once. Keeping this in mind, we devised an algorithm to include the machines sequentially in the analysis. The concept is to add one machine at a time, fixing the sequence (but not the quantities) for the machines solved in previous steps. By doing this, the number of continuous variables increases at each step; however, the number of binary variables remains roughly the same at each step (because we are finding sequences for only the last added machine). We start the algorithm by including first the newer machines, which are usually more cost effective, and thus preferred. At the end, we run a polishing stage; we feed the solution to the solver as a warm-start solution to see if it can improve upon the solution found. Our experience showed that the successive machine inclusion algorithm is very effective at reducing the solution time. When we tested the algorithm with a four-machine instance, we were able to reduce the solution time by more than 600 percent without impacting the solution quality (see Figure 2.)

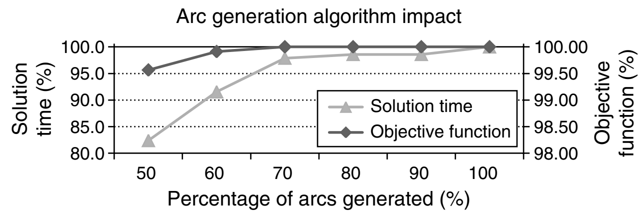

Figure 1: This graph illustrates the impact of using an arc generation algorithm. The solution time and the objective function evolve as the percentage of generated arcs is varied from 50 percent to 100 percent in steps of 10 percent.

Results

Starting from a prototype developed by a Pennsylvania State University team, the Strategic Operations

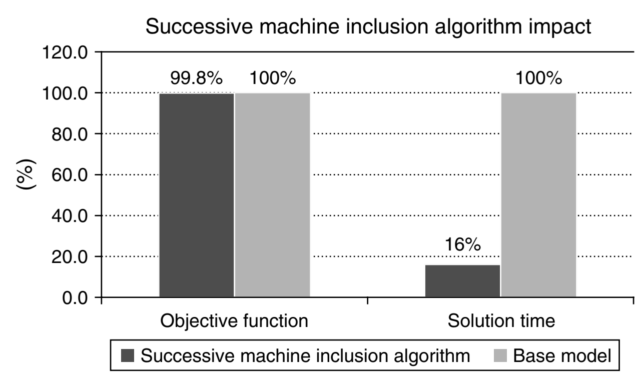

Figure 2: This graph shows a comparison of the objective function and solution time when machines are added sequentially (sequential strategy) and when no special strategy is used (base model). We see that the same solution is obtained in 16 percent of the original time.

Research Team (SORT) at K-C LAO developed an industrial-strength optimization solution using commercial solver software. K-C LAO first used the solution in a pilot plant in January 2008; it then fine-tuned the model parameters to accurately reflect the mill conditions. In the pilot, the solution increased the model's profit function (which is not the total operating profit) by more than 6 percent and provided solutions to the MIP within minutes. K-C LAO has fully deployed the model in its daily operations across multiple business units.

Benefits of the model utilization include planning-time reduction. Developing a traditional plan normally took more than two hours. Currently, the model provides solutions to a similarly sized problem in an average of seven minutes. Furthermore, historical behavior of key business indicators has shown a continuous improvement since the model's implementation. Inventories have been reduced by 45 percent, and service level has improved by 2.4 percent. Figure 3 shows service and inventory-level trends over a 22-month horizon. Although these variables may be altered by other factors, we can conclude that because of continuously using the model, K-C LAO has been able to simultaneously reduce inventory days and increase service levels.

Using the model has allowed the operations team to improve operational efficiency at the plants (see Figure 4). Uptime (i.e., the ratio between production time and total time, including setups and breakdowns) improved 9 percent in 22 months. This increasing uptime trend suggests that more efficient schedules have been developed, although the number of SKUs has increased, which would normally imply an increase in the number of setups.

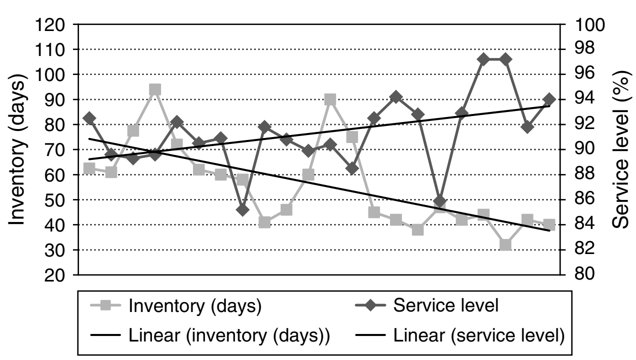

Figure 3: A reduction in inventory level is often associated with a decrease in service level. However, the graph shows that inventories were reduced without affecting service levels. We computed the trend lines using linear regression.

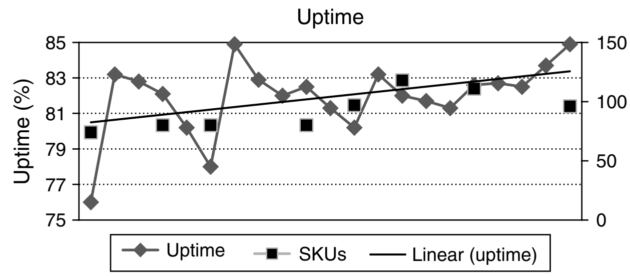

Figure 4: Uptime represents the relationship between production time and total time. A higher value implies better efficiency from the mill point of view because unproductive time is minimized.

Another method that we used to measure the model's impact consisted of comparing the optimal plan suggested by the model (model) to that of the planner (traditional). The results were significant in persuading the production planners to use the model's suggestions (see Tables 1 and 2).

Statistic	Variation: (model − traditional)/traditional (%)
Fill rate (%)	9.9
Income (USD)	6.2
Production cost (USD)	−6.0
Profit (USD)	6.0

Table 1: The table compares the model's results and the traditional plan. Both correspond to Mill 1, August 2009. The fill rate represents the percentage of cases delivered in the same period in which they were requested. The production cost is the variable cost associated with production.

Statistic	Variation: (model − traditional)/traditional (%)
(A) Setup cost (USD)	−17.4
(B) Inventory holding cost (USD)	8.5
(A) + (B)	−13.6
Fill rate	0.0
% machine utilization	−1.0

Table 2: The table compares the model's results and the traditional plan. Both correspond to Mill 2, September 2009.

Tables 1 and 2 show two different comparisons made between the model and traditional plans. Table 1 shows a case in which the fill rate improved. Using the model results, K-C LAO is able to produce and sell more and attain a higher level of revenue. Although this production is achieved at the expense of higher costs, profit is still superior. Table 2 shows a case in which the fill rate in both cases is the same; however, the results proposed by our optimization model suggest a 13 percent model's costs avoidance. Observe that although inventory costs increase by 8.5 percent, setup costs decrease by 17.4 percent.

We added the impacts for four different mills, and analyzed them in terms of the two most important costs: setup and inventory holding costs. To make the comparison as conservative as possible, we forced both plans to produce the exact quantity, so that the changes would only be the result of better machine assignment and sequencing. We achieved an overall reduction of 12 percent in relevant costs; although setup costs were predicted to be 34 percent higher than in traditional plans, only 77 percent of the original holding costs were incurred (see Figure 5).

Conclusions and Ongoing Improvements

In this paper, we discussed K-C LAO's approach to solving a large-scale production-planning and detailed scheduling problem and its impact on planning and operations. The K-C LAO team is currently rolling out the model to five more plants.

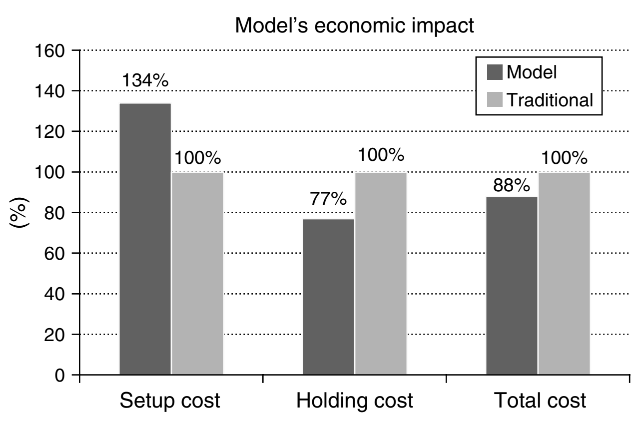

Figure 5: This graph shows the setup and holding costs when comparing traditional and model solutions. Results suggest that total costs will be 12 percent less than using a traditional plan.

The model's impact goes beyond cost reduction, because it allows K-C LAO to quickly run different production plans, and analyze what-if scenarios by comparing the profit function using various target inventories. This can provide insights on which products should be produced in each machine. The model is useful in providing a solution to the planning activities; however, because it can be run throughout the month to suggest modifications of the planning orders, it has also proved to be a dynamic tool, able to adapt to changes in demand and inventories.

K-C LAO management feels comfortable with the model; it has concluded that the model is well calibrated and can be used for policy studies and studying different scenarios. According to K-C Andean Supply Chain Director, Gustavo Palacio, "The importance of the tool developed by PSU and SORT is that planners today can spend more time analyzing scheduling alternatives and see the ramifications of the solutions attained rather than spending time in massaging the data to build one solution. This has allowed us to reduce inventories keeping service levels and better react to the ever-changing market place." To date, K-C LAO has deployed the Web-based decision support system in six plants in five countries and has extended its applicability to production planning, inventory management, and integrated supply chain planning.

Multiple enhancements are also being introduced in the model presented in Appendix A. These include the possibility of keeping stock in multiple locations so that the model is able to produce a gross distribution plan. Also, some machines allow the possibility of packing different SKUs at the same time. Changing packing configurations can be time consuming and costly; the original model was not able to reflect these factors. We are also looking for alternate ways to obtain the end-of-month inventory targets, which depend on the inventory policy and on the sequence to be used the next month. Because sequencing and target inventories depend on each other, we are looking at alternate ways to solve both problems simultaneously (Sánchez et al. 2010). Finally, we are constantly investigating ways to make the model run faster so that longer horizons can be considered.

Appendix A. Model Formulation

In this appendix, we present an MIP formulation of the production lot-sizing and scheduling problem. We first introduce the notation to be used. The model's objective function and constraints are presented and explained in detail.

Sets

$\mathcal{P} = \{1, \ldots, N\}$, Products indexed with i and j.

$\mathcal{M} =$ Machines indexed with m.

$\mathcal{T} = \{1, \ldots, T\}$, Periods indexed with t.

$\mathcal{K} =$ Markets indexed with k. Each market has its own demand and service requirements. This set is used, for example, to distinguish between exports and imports or VIP clients, and so on.

$\mathcal{P}_m =$ Set of products that can be made in machine m. $N_m = |\mathcal{P}_m|$.

$\mathcal{M}_i =$ Set of machines that can produce product i.

$\mathcal{A}_m =$ Set of arcs for transitions allowed for machine m; that is, $\mathcal{A}_m = \{(i, j) \in \mathcal{P}_m \times \mathcal{P}_m$: a transition from product i to j is allowed in machine $m\}$.

$\mathcal{A}^+_{im} = \{j\colon (i, j) \in \mathcal{A}_m\}$. The set of products that can follow product i in machine m.

$\mathcal{A}^-_{im} = \{j\colon (j, i) \in \mathcal{A}_m\}$. The set of products that can precede product i in machine m.

Variables

$x_{imt} =$ Production of product i during period t in machine m, with unit cost c_{im}.

$s_{ikt} =$ Sales volume of product i for market k during period t, with selling price p_{ikt}.

$I_{it} =$ Inventory of i at the end of period t, with unit cost h_i per period. I_{i0} is a known parameter.

$B_{ikt} =$ Back orders of product i for market k at the end of period t, with unit cost b_{ik} per period.

$\tilde{I}^+_{it}, \tilde{I}^-_{it} =$ Overage and shortage inventory with respect to the desired safety stock $\tilde{I}_{it}$, with shortage unit cost $\tilde{b}_i$ per period.

$\delta^{ij}_{mt} =$ 1 if product j is used just after i in machine m during period t with setup cost a_{ij}.

$\gamma^{ij}_{mt} =$ 1 if between periods t and $t+1$ in machine m makes a transition from i and j with setup cost a_{ij}.

$r_{imt} =$ Ranking for product i in machine m in period t. This is a relative ranking; i.e., if there is a transition from i to j, then $r_{jmt} \geq r_{imt} + 1$. If product i is not in the sequence, then this variable is irrelevant.

$y_{imt} =$ 1 if product i is produced in machine m during period t.

$W_{mt} =$ Time that machine m is busy during period t, with unit cost α_{mt}.

Parameters

$D_{ikt} =$ Expected demand for product i, for market k during period t.

$t_{ijm} =$ Sequence-dependent setup time to switch from product i to j in machine m.

$\mu_{im} =$ Product i production rate in machine m.

$K_{mt} =$ Capacity (in hours) for machine m during period t.

$L_i =$ Lower bound for production using product i.

$U_i =$ Upper bound for production using product i (or any big constant if there is no upper bound).

$$\text{Max} \sum_{t\in\mathcal{T},} \sum_{i\in\mathcal{P}} \Bigg[\sum_{k\in\mathcal{K}} \{p_{ikt}s_{ikt} - b_{ik}B_{ikt}\} - \sum_{m\in\mathcal{M}} \{c_{im}x_{imt} + \alpha_{mt}W_{mt}\} - h_i I_{it} - \tilde{b}_i \tilde{I}^-_{it} \Bigg] - \sum_{t\in\mathcal{T}} \sum_{m\in\mathcal{M}} \Bigg[\sum_{(i,j)\in\mathcal{A}_m} a_{ij}\delta^{ij}_{mt} + \sum_{(i,j)\in\mathcal{A}_m} a_{ij}\gamma^{ij}_{mt} \Bigg], \tag{A1}$$

subject to

$$I_{i,t-1} + \sum_{m\in\mathcal{M}_i} x_{imt} - \sum_{k\in\mathcal{K}} s_{ikt} = I_{it} \quad i\in\mathcal{P},\ t\in\mathcal{T}, \tag{A2}$$

$$S_{ikt} + B_{ikt} - B_{ik,t-1} = D_{ikt} \quad i\in\mathcal{P},\ k\in\mathcal{K},\ t\in\mathcal{T} \tag{A3}$$

$$I_{it} - \sum_{k\in\mathcal{K}} B_{ikt} - \tilde{I}^+_{it} + \tilde{I}^-_{it} = \tilde{I}_{it} \quad i\in\mathcal{P},\ t\in\mathcal{T}, \tag{A4}$$

$$\sum_{(i,j)\in\mathcal{A}_m} t_{ijm}\delta^{ij}_{mt} + \sum_{(i,j)\in\mathcal{A}_m} t_{ijm}\gamma^{ij}_{m,t-1} + \sum_{i\in\mathcal{P}_m} \frac{x_{imt}}{\mu_{im}} = W_{mt} \quad m\in\mathcal{M},\ t\in\mathcal{T}, \tag{A5}$$

$$W_{mt} \leq K_{mt} \quad m\in\mathcal{M},\ t\in\mathcal{T}, \tag{A6}$$

$$x_{imt} \geq L_i y_{imt} \quad m\in\mathcal{M},\ t\in\mathcal{T}, \tag{A7}$$

$$x_{imt} \leq U_i y_{imt} \quad i\in\mathcal{P}, m\in\mathcal{M}_i,\ t\in\mathcal{T}, \tag{A8}$$

$$\sum_{j\in\mathcal{A}^-_{im}} \gamma^{ji}_{m,t-1} + \sum_{j\in\mathcal{A}^-_{im}} \delta^{ji}_{mt} - \sum_{j\in\mathcal{A}^+_{im}} \gamma^{ij}_{m,t} - \sum_{j\in\mathcal{A}^+_{im}} \delta^{ij}_{mt} = 0 \quad i\in\mathcal{P},\ m\in\mathcal{M}_i,\ t\in\mathcal{T}, \tag{A9}$$

$$\sum_{j\in\mathcal{A}^+_{im}} \gamma^{ij}_{mt} + \sum_{j\in\mathcal{A}^+_{im}} \delta^{ij}_{mt} = y_{imt} \quad i\in\mathcal{P},\ m\in\mathcal{M},\ t\in\mathcal{T}, \tag{A10}$$

$$\sum_{j\in\mathcal{P}_m} \gamma_{m0}^{i_0 j} = 1 \quad m \in \mathcal{M}, \tag{A11}$$

$$\sum_{i\in\mathcal{P}_m} \gamma_{mT}^{i_0} = 1 \quad m \in \mathcal{M}, \tag{A12}$$

$$r_{jmt} - r_{imt} + N_m(1-\delta_{mt}^{ij}) \geq 1$$
$$(i, j) \in \mathcal{A}_m,\ m \in \mathcal{M},\ t \in \mathcal{T}, \tag{A13}$$

$$x_{imt}, s_{ikt}, W_{mt}, B_{ikt}, r_{imt}, \tilde{I}_{it}^{+}, \tilde{I}_{it}^{-} \in \Re^{+},$$
$$y_{imt}, z_{imt}, \delta_{mt}^{ij}, \gamma_{mt}^{ij} \in \{0, 1\}.$$

The objective function (A1) maximizes the net profit. We compute it as the sales income minus the costs that correspond to backorders, inventory, and machine usage costs. Equation (A2) balances inventory with production and sales. The demand is split according to Equation (A3): it is either sold or increases back orders. Equation (A4) is used to compute shortages ($\tilde{I}_{it}^{-}$) and excess ($\tilde{I}_{it}^{+}$) of net inventory with respect to the desired target inventory ($\tilde{I}_{it}$). Equation (A5) computes the total working time, whereas Equation (A6) bounds it. Equations (A7) and (A8) establish upper and lower bounds to the production batch.

The transitions (i.e., setups) are represented by binary variables δ_{mt}^{ij}, which are 1 if a transition is made between i and j during period t on machine m. Similarly, the variables γ_{mt}^{ij} represent the transitions between the end of one period and the beginning of the next one. Finding a sequence solution on a machine m is equivalent to determining an optimal path (i.e., set of transitions) between the staring node i_0 and the ending node "0" in Figure A.1. If a node is visited, then that SKU can be produced in that period. If a transition is highlighted in the figure, then that transition is used. Equations (A9) and (A10) establish a flow balance for every node, which can be written as

$$y_{imt} = \sum_{j\in\mathcal{A}_{im}^{-}} \gamma_{m,\,t-1}^{ji} + \sum_{j\in\mathcal{A}_{im}^{-}} \delta_{mt}^{ji} = \sum_{j\in\mathcal{A}_{im}^{+}} \gamma_{mt}^{ij} + \sum_{j\in\mathcal{A}_{im}^{+}} \delta_{mt}^{ij}. \tag{A14}$$

This equation says that product i is made at time t if some other product j was made prior to it, either in the same period (in which case $\delta_{mt}^{ji} = 1$) or in the previous period (in which case some $\gamma_{mt-1}^{kj} = 1$). This is equivalent to having path continuity in Figure A.1. Note that Equation (A10) forces y_{imt} to be 1 if the sequence uses product i and simultaneously ensures

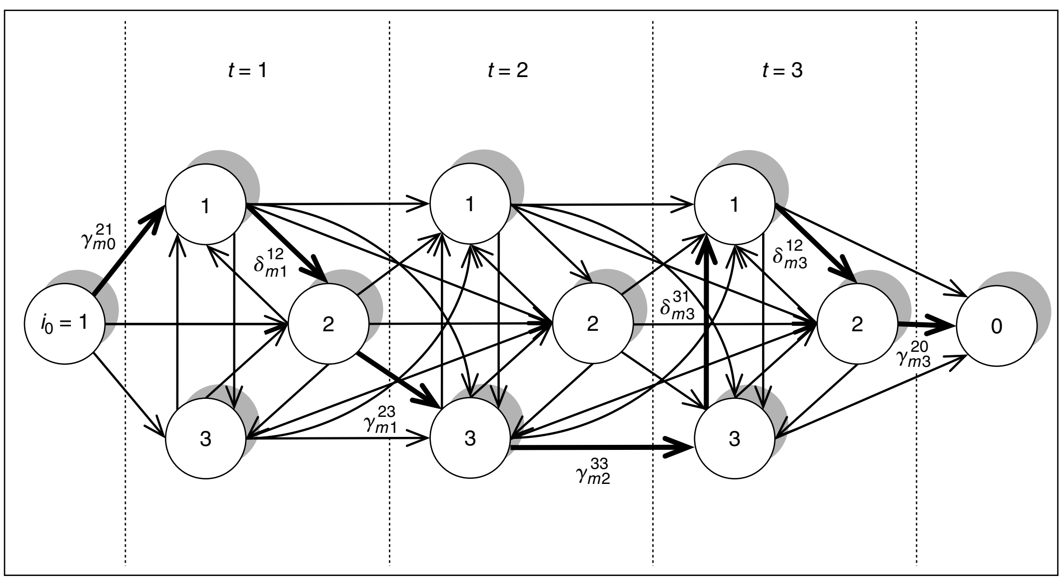

Figure A.1: This graph represents a solution in the scheduling model for a three-product, three-period example. The thin lines represent all possible transitions, whereas the thick lines represent an actual schedule. The solution schedule will be to produce products 1 and 2 in week 1, product 3 in week 2, and products 3, 1, and 2 in week 3. The machine is currently set for product 2.

that at most one δ and γ in each side of Equation (14) is equal to 1. Equation (A11) ensures that the sequence starts with product i_0, whereas Equation (A12) guarantees that the sequence ends with the dummy product at the end of planning horizon T. Finally, Equation (A13) eliminates subtours in the sequence by assigning relative rankings that satisfy $r_{jmt} \geq r_{imt} + 1$ if there is a transition from i to j; otherwise, this restriction gets disabled because the difference in rankings is never bigger than N_m.

We note that Equation (A7) is not establishing a minimum bound for the run, but rather a minimum batch for the period t. If a production run covers more than one period, then we must consider this, and disable this constraint. Thus, Equation (A7) must be replaced by a pair such as

$$x_{imt} \geq L_i(y_{imt} - \gamma^{ii}_{m,t-1} - \gamma^{ii}_{mt}),$$
$$x_{imt} + x_{im,t-1} \geq L_i \gamma^{ii}_{m,t-1}.$$

Notice that the first equation guarantees that if product i is made both in t and $t-1$ or $t+1$, then the bound is disabled, whereas the second equation guarantees that if product i is made in both t and $t-1$ (i.e., $\gamma^{ii}_{m,t-1} = 1$), then the lower bound is active for the sum of production of these two periods. (The previous equations will need straightforward modifications for $t = 1$ and $t = T$.)

Appendix B. Mathematical Descriptions of the Heuristics

In this appendix, we present details about the heuristic algorithms used to speed up the solution.

Assignment Algorithm

The assignment model is a reduced version of the mathematical formulation found in Appendix A without the sequencing constraints; that is, it uses Equation (A1) as the objective, subject to Equations (A2)–(A8) as constraints, setting $\delta^{ij}_{mt} = 0$ and $\gamma^{ij}_{mt} = 0$ in Equation (A5). Call x_{imt} the solution from this assignment. We use it to redefine the new capabilities as

$$\mathcal{P}'_m = \{i \in \mathcal{P}_m : x_{imt} > 0\}.$$

Arc Generation Algorithm

The arc generation algorithm refines the sets of possible arcs $\mathcal{A}_{mt}$ and $\mathcal{B}_{mt}$, thus reducing the number of binary variables. The arc generation algorithm uses deviations from the "optimal sequence," obtained by solving a TSP. Let N_m be the number of SKUs assignable to machine m, as obtained from the assignment algorithm. The arc generation algorithm begins by assuming that, in a given period, all N_m products must be sequenced; it then solves a TSP using an MIP solver to sequence the N_m SKU on that machine. Each SKU is seen to be equivalent of a "city" and the cost associated with setting up the machine from one SKU to the next is the corresponding "distance." The resulting solution is an "optimal sequence."

Let R_{im} be the relative rank of product i in the optimal sequence for machine m. Notice that we can arbitrarily label any SKU as having rank 1 because we have a tour rather than a path. We can define a *cyclic distance* between SKUs i and j as

$$d^m_{ij} = (R_{jm} - R_{im} + N_m) \bmod N_m,$$

where "mod" is the modulus operator. For example, in Figure B.1, the optimal TSP is represented by the thick line that goes clockwise; $R_{1m} = 1$, $R_{3m} = 2$, and so on. The cyclic distance between 1 and 4, d^m_{14}, is 2, whereas the cyclic distance between 5 and 4, d^m_{54}, is 3.

Next, by assuming D_m (a reference maximum cyclic distance) as given, the arc generation approach generates arcs that connect all nodes that are less than D_m distance away from the reference node. Thus, if we are at node 3 and the reference maximum cyclic distance is 2, then generate arcs that connect node 3 to nodes 4 and 2. Mathematically, if (i, j) represents an arc between i and j, then redefine the set of available arcs as

$$\mathcal{A}'_m = \{(i, j) : i, j \in \mathcal{P}_m,\ d^m_{ij} \leq D_m\}.$$

Note that the optimal sequence is generated infrequently (i.e., once every two–three months—only when new products are introduced), and the model uses the stored optimal sequence to generate the set $\mathcal{A}_m$ for developing production plans and detailed schedules. The need to solve the TSPs each time the model is run is, therefore, eliminated.

Successive Machine Inclusion Algorithm

The algorithm can be described as follows: let us assume that machines will be included in the order

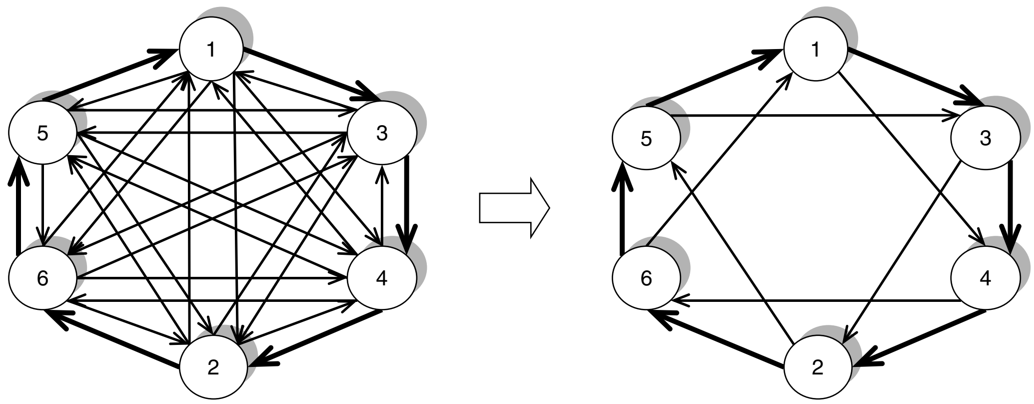

Figure B.1: In this graph, we show the strategy to eliminate arcs. On the left, we see a six-product network in which all transitions are enabled; we see the optimal TSP tour in bold. On the right, we show that we have eliminated all arcs except those that have maximum cyclic distance.

$1, 2, \ldots, M$, where M is the total number of machines. That is, we numerate the machines in the preferred order; therefore, the first machines are those that are more cost effective. At the nth step of the algorithm, the set of machines $\mathcal{M}$ is split as $\mathcal{M}' \cup \{n\}$, where $\mathcal{M}'$ is the set of machines that have the sequence already fixed and n is the machine currently being added to the analysis. Therefore, we will solve the problem described in Appendix A at every step, adding the equations

$$\delta_{mt}^{ij} = \hat{\delta}_{mt}^{ij} \quad m \in \mathcal{M}',\ t \in \mathcal{T},\ (i, j) \in \mathcal{A}_m, \tag{B1}$$

$$y_{imt} = \hat{y}_{imt} \quad m \in \mathcal{M}',\ t \in \mathcal{T},\ i \in \mathcal{P}_m, \tag{B2}$$

where $\hat{\delta}_{mt}^{ij}$ and $\hat{y}_{imt}$ define the fixed sequence found in the solution from previous steps. Notice that fixing the δs and ys automatically fixes the γs.

Acknowledgments

Many people participated in this project at different stages. It would be impossible to mention all of them, but we especially thank LAO Group President Juan Ernesto de Bedout, LAO Supply Chain Vice President Horacio Molas, Andean Region Supply Chain Director Gustavo Palacio, LAO Directors Craig Downing and Guillermo Pinochet, local team members Elcías Millán, Andrés Angel, Ingrid Buelvas, Daniel Bernal, and Andrés Barrera, and previous SORT team members Carlos Moreno, Rodrigo Cruz, Feliza Preciado, Esteban Guardia, and Johanna Lee.

References

Afentakis, P., B. Gavish. 1986. Optimal lot-sizing algorithms for complex product structures. *Oper. Res.* **34**(2) 237–249.

Belvaux, G., L. A. Wolsey. 2000. bc-prod: A specialized branch-and-cut system for lot-sizing problems. *Management Sci.* **46**(5) 724–738.

Belvaux, G., L. A. Wolsey. 2001. Modelling practical lot-sizing problems as mixed-integer programs. *Management Sci.* **47**(7) 993–1007.

Bollapragada, R., U. Rao. 1999. Single-stage resource allocation and economic lot scheduling on multiple, nonidentical production lines. *Management Sci.* **45**(6) 889–904.

Carreno, J. J. 1990. Economic lot scheduling for multiple products on parallel identical processors. *Management Sci.* **36**(3) 348–358.

Cordier, C., H. Marchand, R. Laundry, L. A. Wolsey. 1999. bc-opt: A branch-and-cut code for mixed-integer programs. *Math. Programming* **86**(2) 335–354.

Degraeve, Z., R. Jans. 2007. A new Dantzig-Wolfe reformulation and branch-and-price algorithm for the capacitated lot-sizing problem with setup times. *Oper. Res.* **55**(5) 909–920.

Denton, B. T., J. Forrest, R. J. Milne. 2006. IBM solves a mixed-integer program to optimize its semiconductor supply chain. *Interfaces* **36**(5) 386–399.

Diaby, M., H. C. Bahl, M. H. Karwan, S. Zionts. 1992. Capacitated lot-sizing and scheduling by Lagrangian relaxation. *Eur. J. Oper. Res.* **59**(3) 444–458.

Dixon, P. S., E. A. Silver. 1981. A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem. *J. Oper. Management* **2**(1) 23–39.

Elmaghraby, S. E. 1978. The economic lot scheduling problem (ELSP): Review and extensions. *Management Sci.* **24**(6) 587–598.

Hsu, W. 1983. On the general feasibility test of scheduling lot sizes for several products on one machine. *Management Sci.* **29**(1) 93–105.

Jans, R., Z. Degraeve. 2007. Meta-heuristics for dynamic lot sizing: A review and comparison of solution approaches. *Eur. J. Oper. Res.* **177**(3) 1855–1875.

Jans, R., Z. Degraeve. 2008. Modeling industrial lot sizing problems: A review. *Internat. J. Production Res.* **46**(6) 1619–1643.

Nemhauser, G. L., M. W. P. Savelsbergh, G. C. Sigismondi. 1994. MINTO, a mixed integer optimizer. *Oper. Res. Lett.* **15**(1) 47–58.

Pesenti, R., W. Ukovich. 2003. Economic lot scheduling on multiple production lines with resource constraints. *Internat. J. Production Econom.* **81**(1) 469–481.

Potts, C. N., L. N. Van Wassenhove. 1992. Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. *J. Oper. Res. Soc.* **43**(5) 395–406.

Sánchez, D. R., G. Mejía, G. Riaño. 2010. Long term production planning at Kimberly-Clark Corporation. Working paper, Universidad de los Andes, Bogotá, Colombia.

Simpson, N. C., S. S. Erengüç. 1998. Improved heuristic methods for multiple stage production. *Planning Comput. OR* **25**(7–8) 611–623.

Vanderbeck, F. 1999. Computational study of a column generation algorithm for bin packing and cutting stock problems. *Math. Programming* **86**(3) 565–594.

Wolsey, L. A. 2002. Solving multi-item lot-sizing problems with an MIP solver using classification and reformulation. *Management Sci.* **48**(12) 1587–1602.