Lot-Sizing and Sequencing Optimisation at an Animal-Feed Plant.

to be submitted to Computers and Industrial Engineering

Eli Angela Victor Toso
Reinaldo Morabito

Universidade Federal de São Carlos, Departamento de Engenharia de Produção
São Carlos, São Paulo state, 13565-905, Brazil.
Email: Morabito@power.ufscar.br

Alistair R. Clark*

University of the West of England
Faculty of Computing, Engineering and Mathematical Sciences
Frenchay Campus, Bristol, BS16 1QY, England.
* Corresponding author: Email: Alistair.Clark@uwe.ac.uk

October 13, 2007

Abstract

This paper applies a mixed integer programming model for joint lot sizing and scheduling to a plant for animal feed compounds. A key characteristic of this industry is that certain products can perform a production line cleaning function if a sufficiently large lot is produced between two products that would otherwise require a cleaning setup. Thus the sequence-dependent setup times do not always obey the triangular inequality. Tested on data from the plant, the model takes too long to solve exactly and so several alternative formulations and methods are developed to solve the model more quickly, based on two variants of the Relax and Fix heuristic. The results confirm that the formulations are computationally effective and able to take economic advantage of the intermediate cleaning products. The model schedule substantially improves on that practiced at the feed plant.

Key Words: Lot sizing and sequencing, sequence-dependent setup times, triangular inequality, Relax and Fix heuristic, animal feed production.
1 Introduction

In a manufacturing system, many products often share valuable capacity which is wasted and not used productively when setting up (changing over) from one product to another. Although automation and process engineering has often reduced the magnitude of setups, a large number of companies still face substantial production setup costs and times within an increasing range of products, with consequent losses of production capacity and missed deadlines if setups are not well managed and controlled. Weak performance in this area generally results in backlogs of unmet demand, customer dissatisfaction and loss of company competitiveness.

While many production lots or batches correspond to specific orders and so have a predetermined size, a product or part may instead feed into many small distinct orders with different deadlines. In such a situation, it makes sense to relate the product or part’s lot-sizes to its total demand aggregated from the different orders. In other words, the problem becomes one of simultaneous scheduling and sizing of production lots or batches, based on forecasts of product orders and demand, often under limited production capacity (Askin and Standridge; 1993).

This paper investigates this challenge at Anifeed, a Brazilian animal feed compound company (whose real name has been altered to protect its identity). A mixed integer programming (MIP) model for joint lot sizing and scheduling with sequence-dependent setup times is applied, taking into account that the setup times, like those in many feed plants, do not always obey the triangular inequality. Tests on Anifeed data indicate that the model takes too long to solve exactly and so several alternative formulations and methods are developed to accelerate solution time, making use of the Relax and Fix methods (Wolsey; 1998) on the integer lot-sizes or binary setup variables over time. Computational test results confirm that the Relax and Fix solution acceleration is effective while maintaining quality. The solutions show that the model is able to take advantage of the cleaning function that certain intermediate products can perform if a sufficiently large lot is produced between two products that would otherwise require a cleaning setup. As hoped, the model’s schedules showed a very marked improvement over the schedules implemented at Anifeed.

The rest of this section describes Anifeed’s production process and its scheduling context. Section 2 reviews previous research while section 3 proposes and explains the optimisation model. Section 4 develops alternative solutions methods which are then tested and analysed in section 5 and compared to Anifeeds practice in section 6. Finally section 7 concludes and points out future directions for research.

1.1 The Anifeed production process

Anifeed produces about 200 animal feed supplements which can be grouped into approximately 20 product families. Products within the same family do not contaminate each other and have the same production time per batch. All animal feed supplements follow the same basic production route, and make use of the same key resources: silos, dosing machines, pre-mix machines, mixer, and post-mix packaging, as shown in Figure 1.

Place Figure 1 somewhere around here

The first stage in the production process is to weigh the raw materials based on pre-established formulations. After the operator has specified the feed product and number of batches to be produced, appropriate quantities of the bulkier raw materials are automatically released from the silos into the dosing machines and then held as pre-mix in-process inventory. The bulk materials are transferred to the mixer only after they are all ready
in the pre-mixers. Less bulky materials are stored in bags from which they are manually weighed and added directly to the mixer. Mixing occurs in three phases: dry mixing, addition of fluids, final mixing. The mix is then unloaded into the post-mixer and subsequently bagged. The amount of time spent at these operations varies between product families.

A certain number of batches of each product is made before changing to another product. Each batch measures about 2000 litres, the capacity of the mixer. The amount produced in the mixer depends on the product density, for example: Basic feeds: 2000kg, Premixes: 1440 kg, Mineral Salts: 2400 kg. Technically, the mixer must be at least half-full to ensure efficient mixing, but economically it makes sense to produce a full batch. However, small batches can be produced in the micro-ingredient mixer, and so small orders are not considered in the planning and scheduling of feed supplements.

Further details are available in Toso (2003).

1.2 Production Planning and Scheduling

Although the process has several stages, logically it can be considered to be single-stage given that the stages are arranged serially, the batch flow is continuous and there is no in-process inventory. The capacity bottleneck is the mixer, so it is at this stage that the whole process can be modelled as a one-machine problem, taking total process times into account.

Product changes are frequent, typically about 30 to 40 per week. A complicating feature of the animal-feed industry is that some feed families contaminate others if produced in successive batches. As a result the mixer must be often cleaned, consuming scarce production time. Products in the same family have negligible changeover times, and identical batch weights & processing times. Thus the amount of mixer cleaning time can be minimised by good sequencing of the production of families.

Furthermore, most of Anifeed’s products follow a seasonal pattern of demand, with peaks in certain months. Since employee turnover is high, manpower levels can be adjusted to cope with this seasonality, thus determining basic (pre-overtime) production capacity. The demand in a particular period often exceeds basic capacity, and so overtime is frequently worked to satisfy demand without shortages. When generating production schedules the production manager is particularly concerned to balance overtime and inventory costs, while fulfilling demand without backlogs. The animal feed market is highly competitive, and so delivery delays to clients must be avoided if possible by producing some feed ahead of demand when slack capacity is available.

Thus Anifeed needs to make effective use of production capacity by good lot sizing and sequencing of the production of families. This problem is especially complex for the feed industry due to certain particularities such as highly seasonal demand and sequence-dependent setup times.

2 Review of previous research

At a general level, the lot-sizing problem consists of determining how much to produce of each family or product in every scheduling period, reconciling demand with capacity either through increases in capacity (such as overtime working) or bringing forward production to slacker periods (Johnson and Montgomery; 1974; Hax and Candea; 1984; Trigeiro et al.; 1989; Graves et al.; 1993; Gershwin; 1994). The lot sequencing and scheduling problems consists of determining the order in which to produce lots so as minimise setup costs and/or the setup times that consume productive capacity (Potts and Van Wassenhove; 1992; Lawler et al.; 1993; Allahverdi et al.; 1999; Gupta and Magnusson; 2005).

As noted in the reviews by Drexl and Kimms (1997) and Karimi et al. (2003), lot-sizing and sequencing decisions have often been separately dealt with by many researchers. This
chimes with much industrial practice whereby the sequencing and scheduling of lots is often carried out on the shop floor after the lot sizes have been decided.

However, in the animal feed industry, and in related industries such as soft drinks (Toledo et al.; 2006, 2007), dealing with lot-sizing independently of sequencing makes it difficult to react flexibly to changes in demand and deliver on-time within available capacity. In line with this, the surveys by Drexl and Kimms (1997) and Karimi et al. (2003) show that an increasing number of researchers are jointly considering the two problems of lot sizing and lot-sequencing, and developing a variety of different models and solutions methods, including Salomon et al. (1997); Haase and Kimms (1999); Staggemeier and Clark (2001).

Among the more influential papers on joint lot sizing and scheduling, several stand out in terms of modelling innovation and development. Drexl and Haase (1995) presented the Proportional Lot Sizing and Scheduling Problem (PLSP), which used “small-bucket” time periods during which at most one setup can occur. In contrast, the “large-bucket” representation of the Capacitated Lot Sizing and Scheduling Problem (CLSP) (Haase; 1996) allows many different products per period, but does not sequence the production lots. Fleischmann and Meyr (1997) formulated the so-called General Lot Sizing Problem (GLSP) which models sequence-dependent setup costs on a single machine, allowing multiple setups in each large-bucket time period. Meyr (2000) extended the GLSP to include sequence-dependent setup times, but retained the small-bucket concept, dividing the large planning periods into a predetermined number of micro-periods which contain at most one setup.

Laguna (1999), Clark and Clark (2000), and Clark (2003) also present MIP formulations for lot sizing and sequencing, emphasizing different algorithmic aspects. Laguna (1999) developed a tabu search method with short-term memory while the latter two papers used MIP-based heuristics. Araújo et al. (2007), Toledo et al. (2006) and Luche et al. (2007) present formulations for lot problems in foundries, soft-drinks and electrofused grains production respectively.

The mathematical model initially presented in this paper draws on the General Lot Sizing Problem - Setup Times (GLSP-ST) proposed by Meyr (2000) who considers the loss of capacity resulting from sequence-dependent setup times. It adapts the GLSP-ST by including non-triangular setups times and overtime.

3 Modelling Approach

This paper now presents a MIP model to decide family lot sizes and sequences in each planning period. The aim is to minimise overtime and excess inventory, while satisfying demand and keeping within available capacity. The model results from a combination and adaptation of those in Hax and Candea (1984) and Meyr (2000).

The unit of production is a single batch, whose weight size for a product depends on its density. Anifeed’s sales department aggregates sales orders to result in integer family demand quantities in each period.

To formulate the model, the following indices are used:

\[ i : \text{Product family}, \ i = 1, \ldots, N \]
\[ t : \text{Time period}, \ t = 1, \ldots, T \]
\[ s : \text{Subperiod}, \ s = 1, \ldots, S \]

where:

\[ N = \text{the number of families} \]
\[ T = \text{the number of periods in the planning horizon} \]

\[ S = \text{the total number of sub-periods over the planning horizon} \]

A period \( t \) is split into a fixed number \( S_t \) of subperiods \( s \) of flexible duration (and can even be of zero length). Just one lot (or none) can be produced in each subperiod, so that \( S_t \) is the maximum number of lots that can be produced in a period. Thus, if \( S_t = N \), then all families can be produced in period \( t \) (but do not need to be).

The subperiods do not overlap and the length of each is a decision variable. Subperiods can be viewed as a device to model family changeovers within a period. The length of a subperiod is the sum of the setup time of the family produced within it and the actual production time of its single lot. If no lot is produced, then this duration can be zero in which case the machine’s setup state is conserved. In other words, if family \( i_{s-1} \) is produced in subperiod \( s-1 \) and subperiod \( s \) is inactive, then no setup time is needed at the start of subperiod \( s+1 \) to resume production of family \( i_{s-1} \).

If the same family is produced in consecutive subperiods, then the family lot size is the sum of the production quantities in these subperiods. Thus a lot can be produced over multiple consecutive subperiods and periods. The subperiod decisions determine the number, size and sequence of the family production lots.

The input data required by the model are:

- \( C_t \): Available capacity time in each period \( t \).
- \( p_i \): Time needed to produce one batch of each product family \( i \).
- \( l_{m_i} \): Minimum lot size of family \( i \) (integer number of batches).
- \( h_i \): Cost in monetary units (m.u.) of holding one week’s inventory of family \( i \).
- \( c_{o_t} \): Unit cost of overtime for week \( t \).
- \( st_{ji} \): Setup times needed to changeover from product family \( j \) to family \( i \).
- \( d_{it} \): Forecasts of demand for family \( i \) at the end of week \( t \) in the planning horizon.
- \( I_{i0} \): Inventory of family \( i \) at the start of the planning horizon.
- \( x_{i0} \): indicates (= 1) if production is already set up to produce family \( i \) at the start of the first period (otherwise = 0).
- \( u_t \): Upper limit on the number of overtime hours permitted in period \( t \).

The decisions output by the model will be:

- \( I_{it} \): Inventory of product family \( i \) at the end of period \( t \).
- \( q_{is} \): Number of batches of family \( i \) produced in subperiod \( s \) (integer variable).
- \( x_{is} \): indicates (= 1) if production is to be set up for product family \( i \) in subperiod \( s \) (otherwise = 0).
- \( y_{jis} \): indicates (= 1) if production is to be changed over from product family \( j \) to family \( i \) in subperiod \( s \) (otherwise = 0).
- \( O_t \): Number of overtime hours needed in period \( t \).
A MIP formulation of the problem is:

Minimise \[ \sum_i \sum_t h_{it} I_{it} + \sum_t c_o O_t \]  

such that

\[ I_{it} = I_{i,t-1} + \sum_{s \in S_t} q_{is} - d_{it} \quad \forall i, t \]  

\[ \sum_i \sum_{s \in S_t} p_i q_{is} + \sum_j \sum_i \sum_{s \in S_t} s t_{ji} y_{jis} \leq C_t + O_t \quad \forall t \]  

\[ p_i q_{is} \leq (C_t + u_t) x_{is} \quad \forall i, t, s \in S_t \]  

\[ \sum_i x_{is} = 1 \quad \forall s \]  

\[ y_{ijis} \geq x_{i,s-1} + x_{js} - 1 \quad \forall i, j, s \]  

\[ q_{is} \geq lt_i(x_{is} - x_{i,s-1}) \quad \forall i, s \]  

\[ 0 \leq O_t \leq u_t \quad \forall t \]  

\[ I_{it} \geq 0 \quad \forall i, t \]  

\[ q_{is} \geq 0 \text{ and integer} \quad \forall i, s \]  

\[ x_{is} \in \{0, 1\} \quad \forall i, s \]  

\[ 0 \leq y_{ijis} \leq 1 \quad \forall i, j, s \]  

where the symbol ∃ means ‘for all’.

The objective function (1) minimizes the costs of inventory and overtime, two criteria of major importance to the company. Differently to Hax and Candea (1984) and Meyr (2000), it does not consider setup costs, the reason being that setups basically consume just labour and time, neither of which incurs an immediate direct cost. Manpower levels are fixed over the planning horizon. Setup times are not directly penalized in the objective function (1), but rather indirectly though their use of overtime. Thus the minimisation of overtime in the objective function will prevent superfluous setups via constraints (3). The costs of shortages or backlogs are not included in (1) as the company is always able to avoid these through the use of overtime.

Constraints (2) balance inventory, production and demand over consecutive periods. The capacity constraints (3) take into account the setup times as well as actual production times, and the possibility of a limited amount of overtime. Note that if family \( i \) is produced in subperiod \( s \), then the duration of this subperiod is the setup time \( st_{ji} \) \( y_{jis} \) needed to changeover from the previous family \( j \) plus the production time \( p_i q_{is} \), as illustrated in Figure 2.

**Place Figure 2 somewhere around here**

Constraints (4) ensure that production of a family can occur in a subperiod only if the line is set up accordingly. Constraints (5) restricts setups in any subperiod to just one family. Constraints 6) relate setup states to setup changeovers, i.e., if production changes from
family $i$ in subperiod $s-1$ to family $j$ in subperiod $s$ then the binary changeover variable $y_{ijs}$ must have value 1.

In constraints (7), the minimum lot-size is enforced if family $i$ was not produced in the previous subperiod. A minimum lot size is needed as setup times do not always satisfy the triangular inequality. To better understand why, consider a family $i$ whose production contaminates that of family $k$ unless a thorough cleaning occurs as part of the substantial setup time $st_{ik}$. However, in the animal feed industry, such cleaning can sometimes occur during the production of an intermediate family $j$ whose setup time $st_{ij}$ from $i$ and setup time $st_{jk}$ to $j$ are both shortened to the extent that $st_{ik} > st_{ij} + st_{jk}$. Thus the triangular inequality $st_{ik} \leq st_{ij} + st_{jk}$ does not hold in this case. Without constraints (7) to impose sufficient production of $j$ to allow proper cleaning of $i$’s contaminants, an optimal schedule could setup from $i$ to $k$ via zero production of $j$ rather than directly.

Constraints (8) impose limits on overtime working, and constraints (9) prohibit backlogs of demand. Constraints (10) require the production to be a whole number of batches while constraints (11) ensure the setup state variables $x_{is}$ to be binary.

The changeover $y_{ijs}$ variables will turn out to be binary in an optimal solution without having to impose this in constraints (12), as demonstrated by the following *reductio ad absurdum* argument: if there were an optimal solution with fractional $y_{ijs}$ values, then constraints (3) and (6) imply that it is always possible to find a better solution by reducing the value of $y_{ijs}$ to be 0 when $0 < y_{ijs} < 1$ and to remain at 1 when $y_{ijs} = 1$. However, the absence or presence of constraints (12) does make a difference when the model is used within a heuristic procedure to obtain near-optimal solution, as shown in section 5 below.

Model (1) to (12) has $N(NS + 2S + T) + T + 1$ variables, of which $NS$ are binary, and $N$ is integer. If the variables $y_{ijs}$ were modelled as binary, then there would be $NS(N+1)$ binary variables, i.e., $N+1$ times more.

## 4 Solution Methods

The model above was implemented in the GAMS mathematical programming language (Brooke et al.; 1992) and version 7.0 of the Cplex optimisation solver. (Ilog; 2000). The tests were run on an Intel Dual Centrino 1664 Mhz processor with 2Gb of RAM under Windows XP.

The test data from Anifeed was suitably distorted to maintain confidentiality but retain proportionality. The data initially comprised two 4-week months of production, corresponding to the Brazilian rainy and dry seasons. Compared to the dry season, demand in the rainy season is spread over a larger number, 21, of product families, albeit each with smaller demand. This demand varied substantially between families - a few had zero or very small demand over the four weeks while just five families accounted for three quarters of all demand, as illustrated in Figure 3. Demand varied also over time within each family and, in total, would require overtime in week 4 if a lot-for-lot production policy were followed.

**Place Figure 3 somewhere around here**

Family changeover times $st_{ji}$ were either 100 minutes or zero (rounded down from negligible near-zero times). The family production times $p_i$ and inventory holding costs $h_i$ ranged between 0.2 & 0.6 hours per batch and 102 & 922 monetary units per period respectively (the exact data set is available from the authors). The capacity $C_t$, overtime limit $u_t$ and overtime costs were constant and equal to 64 hours, 16 hours and 859.2 monetary units per hour respectively.
Formulation (1) to (12) had 40,661 variables (of which 1,764 were integer and 1,764 binary) and 40,749 constraints. Initial computational tests showed that there was little improvement in the incumbent solution after several hours of CPU time, with a very large associated duality gap in relation to the Cplex lower bound. Several ways of improving the solution and computing time were investigated as now explained.

The following constraints were proposed by Fleischmann and Meyr (1997) as valid inequalities for a GLSP-ST model in order to reduce the search space:

\[
\sum_{j \neq i} y_{ji,s} - 1 \geq \sum_{j \neq i} y_{jis} \quad \forall t, s = f_t + 2, ..., l_t
\]

(13)

\[
q_{is} \leq \frac{C_t + u_t}{p_t} \left(2 - \sum_{j} y_{ji,s-1} - y_{iis}\right) \quad \forall i, t, s = f_t + 1, ..., l_t
\]

(14)

where \(f_t\) and \(l_t\) are respectively the first and last subperiods of period \(t\). Within a period, constraints (13) ensure that setups which involve a change of family (i.e., \(\sum_{j \neq i} y_{ji,s} = 1\)) are carried out before those which do not (i.e., \(\sum_{j \neq i} y_{jis} = 0\)). Constraints (14) prevent the production of the same family in two consecutive subperiods within a period. However the computational results for the company data reported below show that the incumbent solution after 1 hour did not improve.

Close examination of the incumbent solution revealed that some of its \(y_{jis}\) changeover variables had value 1 when the corresponding production quantity \(q_{is}\) was 0. The formulation permits this and it occurred particularly for the zero changeover times. In an attempt to flush out these misleading phantom setups and hopefully speed up convergence to an optimal solution, the \(y_{jis}\) variables were penalized in the objective function, but to no effect. We also added the redundant constraints \(\sum_{i,j} y_{ijs} = 1\) (given that \(st_{ii} = 0\) \(\forall i\)), but again to no effect.

We also tested including the constraints (15) to avoid a changeover to a family (\(y_{jis} > 0\)) when it is not then produced (\(x_{is} = 0\)), as allowed without setup cost by the objective function. In other words, variable \(y\) is free to be positive when it does not need to be and could just as well be zero. The aim of including constraints (15) was to reduce the search space without loss of generality in an attempt to improve the solution quality within a fixed amount of computing time.

\[
\sum_{j} y_{jis} \leq x_{is} \quad \forall i, s
\]

(15)

Again the tests reported below for the company data showed that it was largely ineffective in obtaining a better solution within a hour’s CPU time.

In order to accelerate the solution time, three variants of the Relax and Fix heuristic were developed and tested. The Relax and Fix method (Wolsey; 1998) solves a series of partially relaxed MIPs, each with a number of integer variables that is small enough to be quickly and optimally solved by Cplex’s branch and cut default algorithm. As the series progresses, each set of integer variables is permanently fixed at their solution values, and the relaxed variables are reduced in number, eventually disappearing. The procedure is broadly similar to a depth-first identification of an initial integer solution for a MIP model in a branch and bound search. Its big advantage is its speed.

The three variants of the Relax and Fix heuristic were:

1. **Method RF on \(q_{is}\):** Relax and Fix on the \(q_{is}\) variables: Relax only integrality of the \(q_{is}\) variables, solve model (1) to (12), then fix the values of the \(x_{is}\) and \(y_{jis}\) variables and resolving with the \(q_{is}\) variables constrained to be integer. The rationale is that the relaxed solution will be near the optimal solution, but there is a possibility that the procedure might not reach a feasible solution.
2. Method RF on \( t \) Forwards: Relax and Fix forward in time: Maintaining the \( q_{is} \) variables as integer, relax the \( x_{is} \) and \( y_{jis} \) variables in periods 2 onwards, solve (1) to (12), permanently fix the period 1 solution values, restore integrality constraints to the \( x_{is} \) and \( y_{jis} \) variables in period 2 but relax them for periods 3 onwards, solve (1) to (12) again, permanently fix the period 2 solution values, restore integrality constraints to the \( x_{is} \) and \( y_{jis} \) variables in period 3 but relax them for periods 4 onwards, solve (1) to (12) again, and so on until period \( T \).

3. Method RF on \( t \) Backwards: Relax and Fix backward in time: As immediately above, but backwards in time over periods \( t = T, T-1,...,1 \).

All the above three variants of the Relax and Fix heuristic are tested in the next section.

5 Computational Tests and Results

The purpose of the tests was to gain insight rather than to carry out conclusive experimentation. However the use of real operational four-week data from Anifeed enabled a comparison with the company’s own scheduling. It also provided a basis for further research.

The test results for both the initial rainy and dry season data sets are shown in Table 1. The two columns for each solution shows its cost (objective function value) and the computational time when it was identified as an incumbent (limited to a maximum total of one hour for all methods, i.e., the 4-period Relax and Fix backwards/forwards methods had a limit of 15 minutes for each period’s MIP).

For the rainy season, note that method RF on \( q_{is} \) performs best, having a cost of 3,453. The best incumbent solution for the initial relaxed problem took 30 minutes to be identified within the total time limit of 60 minutes that was considered a reasonable upper time limit for the decisions involved. After this, the final integer solution then took just one second to be found. This solution may or may not be optimal, but we do know that its cost is at most 14% higher that of the optimal solution. Its lot sizes and sequence are shown in Table 2. The “empty” subperiods, for example, subperiods 3 and 4, are of zero length and simply indicate that less than \( S_1 \) \((= N = 21)\) lots were produced. Additional constraints can force such empty subperiods to occur at the end of each period, but cause extra computational burden and so were not used.

Examining Table 2, note that demand is totally satisfied by production in period 1, leaving over enough capacity to produce for stock 1 batch of family 2, 2 batches of family 19 and 3 of family 21. Table 2 shows that two non-trivial setups to families 20 and 19 were required, so that total required capacity was 63.14 hours (59.9 for production and 3.34 for setups).

In period 2, demand is met by production except for family 19 which calls on one batch of the stock produced in period 1 leaving one batch to carry over to period 3, i.e., \( I_{19,2} = 1 \). There is still enough capacity left over to produce for stock 1 batch of family 19, 5 batches of family 20 and 11 of family 21, giving \( I_{9,2} = 2, I_{5,2} = 20 \) and \( I_{21,2} = 14 \). Note from Table 2 that the production needs just one setup, requiring 63.97 hours in all.

In period 3, demand is met by production except again for family 19 which calls on the remaining one batch of stock and four batches of the stock of family 20, leaving one batch
to carry over to period 4, i.e., $I_{19,3} = 0$ and $I_{20,3} = 1$. A further 19 batches of family 21 are produced for stock, thus using up all 64 hours of capacity without a single setup.

Finally, in period 4, demand exceeds capacity and is met with the stock produced earlier. Again, all 64 hours of capacity are used without a single setup.

All four periods avoided the use of overtime, so the solution value of 3,453 m.u. is composed entirely of stock costs. Observe from Table 2 that the changeover from family 21 to family 9 (underlined) between subperiods 43 and 44 performs a contaminant cleaning function, given that the triangular inequalities is not obeyed for setup times in the family sequence $21 \rightarrow 9 \rightarrow 4$. A direct changeover from family 21 to family 4 would have required a non-trivial cleaning setup.

Returning to Table 1, observe that the RF on $q_{is}$ method also performs best (and fastest) for the dry season data. The basic model (1) to (12) provides a reasonable solution for both seasons, while the RF on $t$ methods perform well in different seasons.

To further explore the relative performance of the models and methods, additional tests were run on each of 7 consecutive months the following year. The results are reported in Table 3. Note the superior performance of the RF on $q_{is}$ method.

6 Comparison with Anifeed’s own plan

Anifeed follows a chase strategy (Nahmias; 1995) of producing only the demand forecast for the forthcoming week, making substantial use of overtime. Sequencing of production is not planned, but rather carried out on the shop floor. For example, production records for the same rainy season month in hand show that there were 4 non-trivial setups in period 1, 3 setups in period 2, 4 in period 3, and 2 in period 4. This required total production and setup times of 64.18, 61.41, 67.98 and 78.25 hours respectively over the 4 periods, giving an objective function (1) of 15,809 m.u. compared to best solution of 3453 m.u. in Table 1, giving the model a striking advantage of 78%.

During the dry season, characterized by a smaller variety of products, but each with much larger demand, the model solutions are more uniform and generally faster. The planning is also simpler for Anifeed, and the 8% advantage of the model was much smaller than for the rainy season.

Tests over the same 7 months as Table 3 showed that the model reduced costs by 31% compared to the best possible solution that Anifeed could achieve using its chase strategy. Thus the objective function value was substantially reduced, due to the release of capacity as a result of better sequencing and thus the anticipation of production in slack periods.

Even without improved sequencing, the gap between the Anifeed and model results could have been reduced if Anifeed had not followed its chase strategy and instead had brought production forward from periods 3 and 4 to take advantage of the underuse of capacity in period 2. However, the company is wary of over-producing to demand forecasts that will invariably change. One way around this would be to adopt a rolling horizon strategy of implementing only the immediate period’s production schedule and then to reschedule the following period’s production with updated forecasts of demand, as in Clark and Clark (2000).

7 Conclusions and Future Research

This paper developed a mixed integer programming model for joint lot sizing and scheduling with sequence-dependent setup times, motivated by an animal feed plant where such
times do not obey the triangular inequality. Tested on data from the plant, the exact model
takes too long to solve exactly and so several alternative formulations and methods are
explored to solve the model more quickly. The method that seems most promising is Relax
and Fix (RF) on the integer lot-sizes $q_{is}$.

The results confirm that the model is able to take advantage of the "cleaning" function
that certain intermediate product families can perform if a sufficiently large lot is produced
between two families that would otherwise require a non-trivial setup. The model solution
is a clear improvement on that practiced at the feed plant, but further testing of the RF on
$q_{is}$ and other methods to properly compare them with Anifeed’s practice, particularly on
the basis of rolling horizon usage.

A continuing challenge is to develop exact and approximate solution approaches that
are much faster yet guarantee near-optimal solutions. To this end, research directions that
the authors are developing include (i) methods for model decomposition, constraint relax-
ation, including further development of relax-and-fix based heuristics (Clark; 2003); (ii) lot
sequencing methods based on the Asymmetric Travelling Salesman problem (ATSP) meth-
ods that very efficiently solve a series of Assignment Problems with sub-tour elimination
constraints (Lawler et al.; 1985); (iii) alternative optimisation approaches based on modern
metaheuristics such as memetic algorithms (Hart et al.; 2005; Krasnogor and Smith; 2005;
Toledo et al.; 2007).
References


### Table 1: Computational Test Results for Dry and Rainy seasons

<table>
<thead>
<tr>
<th>Approach</th>
<th>Rainy season</th>
<th>Dry season</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obj. Value</td>
<td>Time (secs)</td>
</tr>
<tr>
<td>(1) to (12)</td>
<td>3,931</td>
<td>1305</td>
</tr>
<tr>
<td>(1) to (14)</td>
<td>11,270</td>
<td>1305</td>
</tr>
<tr>
<td>(1) to (12), (15)</td>
<td>17,166</td>
<td>1160</td>
</tr>
<tr>
<td>(1) to (15)</td>
<td>11,367</td>
<td>791</td>
</tr>
<tr>
<td>RF on $q_{is}$</td>
<td>3,453</td>
<td>1800</td>
</tr>
<tr>
<td>RF on $t$, Forward</td>
<td>Infeasible</td>
<td>na</td>
</tr>
<tr>
<td>RF on $t$, Backward</td>
<td>3,453</td>
<td>na</td>
</tr>
</tbody>
</table>

*na indicates not applicable*
Figure 1: Production Process of Anifeed’s Food Supplements
Figure 2: Setup and production times within Capacity
Figure 3: Family demand over time
<table>
<thead>
<tr>
<th>Period t=1</th>
<th>Period t=2</th>
<th>Period t=3</th>
<th>Period t=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>i</td>
<td>q_{is}</td>
<td>a</td>
</tr>
<tr>
<td>1</td>
<td>f5</td>
<td>25</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>f2</td>
<td>2</td>
<td>23</td>
</tr>
<tr>
<td>3</td>
<td>f20</td>
<td>4</td>
<td>Setup</td>
</tr>
<tr>
<td>4</td>
<td>f12</td>
<td>2</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>f9</td>
<td>1</td>
<td>26</td>
</tr>
<tr>
<td>6</td>
<td>f15</td>
<td>1</td>
<td>27</td>
</tr>
<tr>
<td>7</td>
<td>f8</td>
<td>29</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>f10</td>
<td>58</td>
<td>29</td>
</tr>
<tr>
<td>9</td>
<td>f7</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>f9</td>
<td>40</td>
<td>31</td>
</tr>
<tr>
<td>11</td>
<td>f17</td>
<td>9</td>
<td>32</td>
</tr>
<tr>
<td>12</td>
<td>f14</td>
<td>12</td>
<td>33</td>
</tr>
<tr>
<td>13</td>
<td>f19</td>
<td>2</td>
<td>34</td>
</tr>
<tr>
<td>14</td>
<td>Setup</td>
<td>35</td>
<td>f8</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>36</td>
<td>f15</td>
</tr>
<tr>
<td>16</td>
<td>f11</td>
<td>2</td>
<td>37</td>
</tr>
<tr>
<td>17</td>
<td>f21</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>18</td>
<td>f17</td>
<td>1</td>
<td>39</td>
</tr>
<tr>
<td>19</td>
<td>f3</td>
<td>9</td>
<td>40</td>
</tr>
<tr>
<td>20</td>
<td>f16</td>
<td>1</td>
<td>41</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>42</td>
<td>f21</td>
</tr>
</tbody>
</table>

Table 2: Lot Sizes and Sequence - Rainy Season (RF on q_{is})
<table>
<thead>
<tr>
<th>Approach</th>
<th>month 1</th>
<th>month 2</th>
<th>month 3</th>
<th>month 4</th>
<th>month 5</th>
<th>month 6</th>
<th>month 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) to (12)</td>
<td>4,438</td>
<td>5,159</td>
<td>9,795</td>
<td>3,378</td>
<td>9,886</td>
<td>10,113</td>
<td>12,014</td>
</tr>
<tr>
<td></td>
<td>175</td>
<td>252</td>
<td>893</td>
<td>50</td>
<td>43</td>
<td>56</td>
<td>2,321</td>
</tr>
<tr>
<td>(1) to (14)</td>
<td>3,131</td>
<td>5,304</td>
<td>33,529</td>
<td>3,378</td>
<td>9,905</td>
<td>10,113</td>
<td>15,657</td>
</tr>
<tr>
<td></td>
<td>754</td>
<td>767</td>
<td>2,500</td>
<td>444</td>
<td>1,018</td>
<td>1,081</td>
<td>3,016</td>
</tr>
<tr>
<td>(1) to (12), (15)</td>
<td>3,157</td>
<td>13,298</td>
<td>12,713</td>
<td>3,407</td>
<td>9,887</td>
<td>10,113</td>
<td>16,123</td>
</tr>
<tr>
<td></td>
<td>582</td>
<td>1,233</td>
<td>935</td>
<td>3,085</td>
<td>139</td>
<td>2,334</td>
<td>1,171</td>
</tr>
<tr>
<td>(1) to (15)</td>
<td>3,131</td>
<td>16,441</td>
<td>11,447</td>
<td>3,378</td>
<td>9,887</td>
<td>11,234</td>
<td>33,141</td>
</tr>
<tr>
<td></td>
<td>2,195</td>
<td>1,940</td>
<td>3,600</td>
<td>541</td>
<td>1,530</td>
<td>1,642</td>
<td>1,514</td>
</tr>
<tr>
<td>RF on (q_{ls})</td>
<td>3,096</td>
<td>5,038</td>
<td>10,584</td>
<td>3,378</td>
<td>9,875</td>
<td>10,018</td>
<td>11,219</td>
</tr>
<tr>
<td></td>
<td>138</td>
<td>374</td>
<td>1,800</td>
<td>50</td>
<td>265</td>
<td>134</td>
<td>332</td>
</tr>
<tr>
<td>RF on (t) Forward</td>
<td>3,981</td>
<td>8,236</td>
<td>23,693</td>
<td>3,378</td>
<td>10,200</td>
<td>10,442</td>
<td>11,219</td>
</tr>
<tr>
<td></td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
</tr>
<tr>
<td>RF on (t) Backward</td>
<td>3,131</td>
<td>5,141</td>
<td>10,584</td>
<td>5,284</td>
<td>10,874</td>
<td>10,961</td>
<td>11,622</td>
</tr>
<tr>
<td></td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
</tr>
</tbody>
</table>

Table 3: Computational Test Results for 7 consecutive months
[na indicates not applicable]