Production Lot Sizing and Scheduling with Non-Triangular Sequence-Dependent Setup Times

Submitted to
International Journal of Production Economics

Alistair Clark * and Masoumeh Mahdieh
Department of Engineering Design and Mathematics
University of the West of England, Bristol, BS16 1QY, United Kingdom.

Socorro Rangel
Departamento de Ciências de Computação e Estatística
Universidade Estadual Paulista (Unesp), Sao José do Rio Preto SP, Brazil.

* Corresponding author: `Alistair.Clark@uwe.ac.uk`

October 3, 2012

Abstract

This article considers a production lot sizing and scheduling problem with sequence-dependent setup times that are not triangular. Consider, for example, a product p that contaminates some other product r unless either a decontamination occurs as part of a substantial setup time st_{pr} or there is a third product q that can absorb p's contamination. When setup times are triangular then $st_{pr} \leq st_{pq} + st_{qr}$ and there is always an optimal lot sequence with at most one lot (AM1L) per product per period. However, product q's ability to absorb p's contamination presents a shortcut opportunity and could result in shorter non-triangular setup times such that $st_{pr} > st_{pq} + st_{qr}$. This implies that it can sometimes be optimal for a shortcut product such as q to be produced in more than one lot within the same period, breaking the AM1L assumption in much research. This article formulates and explains a new optimal model that not only permits multiple lots (ML) per product per period, but also prohibits subtours using a polynomial number of constraints rather than an exponential number. Computational tests demonstrate the effectiveness of the ML model, even in the presence of just one decontaminating shortcut product, and its fast speed of solution compared to the equivalent AM1L model.

Key Words: Lot sizing and scheduling, Sequence-dependent setup times, Non-triangular setup times.

1 Introduction

Some manufacturing systems have to meet a regular but varying demand for products. When manufacturing capacity is limited, such demand cannot be met instantaneously from production, but from inventory accumulated previously. Lot-sizing decisions then need to be made about how much of each product to produce in each demand period and how much inventory to accumulate in order to meet demand while keeping within production capacity.

If a setup cost or time is charged to change from one product to another, then a sequence or schedule of lots also needs to be decided. Many setups are sequence-dependent, that is, the size of the setup charge depends on the product processed immediately beforehand. For example, it often takes less time to setup to a similarly-coloured product than to one with a very different colour. Furthermore, such setup times are sometime asymmetric, for example, it may take more time to setup from a dark coloured product to a light one than vice-versa. Such distinctions matter because decisions involving sequence-dependent setup times are generally more complex computationally than ones with sequence-independent times.

Many manufacturers separate lot sizing decisions from lot sequencing in order to simplify the complexity of the decision-making. Often sequences are decided first and then lot sizes are determined taking into account forecasts of demand. However, this can result in production being less effective and more costly than it needs to be. If a product has relatively low demand then producing it frequently is probably not making efficient use of production capacity. It may be more cost effective to produce infrequent lots, economise on setup times and hold some of the product in inventory over several demand periods. But deciding exactly which products should be produced infrequently and in which periods is essentially a lot sizing decision that should be made before or with the sequencing decision. In other words, lot sizing and sequencing decisions are ideally made jointly and simultaneously rather than separately in order to competitively satisfy the demand for products within available production capacity.

This paper formulates and tests a new model for lot sizing and sequencing with asymmetric sequence-dependent setup times, and in particular for non-triangular ones. After a literature review in section 2, the model is developed in section 4 using a polynomial number of multi-commodity-flow-type constraints adapted from Claus (1984), and then computationally tested in section 5. The paper concludes in section 6 with a discussion of the model's value and flags remaining challenges and opportunities for future research.

2 Lot Sizing and Sequencing

Research into production lot sizing and scheduling has progressed substantially over the last decades, as shown in the reviews by Drexl and Kimms (1997) and Karimi et al. (2003), recent research (Kovács et al.; 2009), and a special issue (Clark et al.; 2011). In July 2010 at the *24th European Conference on Operational Research* (EURO) in Lisbon, a stream on lot sizing and scheduling was organized for the first time in the history of this conference, containing seven sessions with more than 25 presentations.

In particular, much progress has been made in the area of lot sequencing when setup times are sequence-dependent (Meyr; 2000, 2002; Clark and Clark; 2000; Araújo et al.; 2007). The General Lot-sizing and Scheduling Problem (GLSP), developed by Fleischmann and Meyr (1997), minimizes inventory and sequence-dependent setup costs on a single machine with finite capacity, allowing multiple setups in each single 'large-bucket' time period. The GLSP was extended by Meyr (2000) to consider sequence-dependent setup times

(GLSP-ST). Toso et al. (2009) reformulated the GLSP-ST model to permit backlogging and non-triangular setup times, but still assumed at most one lot per product in each period.

Clark et al. (2010) pursued an alternative approach via the Asymmetric Travelling Salesman Problem (ATSP) which has been very extensively researched (Lawler et al.; 1985; Carpaneto et al.; 1995). The adaptation of the ATSP to modelling lot-sizing and scheduling with sequence-dependent setups is not direct, since the production system is often already setup for a particular product (that is starting at a given city) and some products might not be produced in a given period if the demand is sufficiently small or the capacity tight (Clark et al.; 2010).

A method that has been found to be successful in practice for optimally solving the ATSP is to quickly solve the corresponding Assignment Problem (AP) as a linear programme, identify the resulting subtours, and then resolve the AP, explicitly prohibiting these subtours using a potentially exponential number of Dantzig-Fulkerson-Johnson-type constraints adapted from Dantzig et al. (1954). The method carries on iteratively in this manner until no subtours result. It can be used heuristically (and its convergence rate sometimes accelerated) by patching the subtours into a single tour at each iteration (Karp; 1979), thus providing a feasible solution (and an upper bound). Clark et al. (2010) adapted the ATSP subtour elimination method to lot sequencing over multiple periods with setup carryover between periods. An extension of the method then used a patching heuristic to accelerate the time to converge to a provably optimal solution.

3 Non-triangular setup times

In certain industries, such as animal feed supplements, some products can contaminate other products. For instance, copper is essential for pigs but kills sheep even in tiny doses. Contamination is a particular concern for the feed industry, although the problem is general and similar concerns also exist in a diverse range of other industries, such as food & beverages, and the oil industry. In the feed industry, blending equipment must be cleaned in order to avoid contamination, resulting in substantial setups that consume scarce production time. Fortunately, the amount of cleaning can be minimised by the effective sequencing of production lots.

Certain intermediate "cleansing" or shortcut products can cause non-triangular setup times. These products clean the machines whilst being processed (for example, certain wheat mixtures) and hence reduce overall setup times. In other words, contamination cleaning can occur during value-adding production time as well as during non-productive setup time.

More precisely, "triangular" sequence-dependent setup times st occur when it is never worse to set up from product p to r directly than to setup via a third product q, so that the triangular inequality of setup times: $st(p,r) \leq st(p,q) + st(q,r)$ always holds. However, in the animal feed and other industries, the contamination of a product r by a previous product p just beforehand can be often avoided by producing enough of an intermediate product q so that it absorbs p's contamination. For this to save time, the triangular inequality must not hold in this case, that is, the sum of the setup times $st(p,q)$ from p and $st(q,r)$ to r must be short enough so that $st(p,q) + st(q,r) < st(p,r)$. Figure 1 shows an example of these two possibilities. The nodes represent products and the arcs indicate possible production sequences. The continuous arcs on the left side of Figure 1 indicate that in the case of triangular setups it is better to change over from product p directly to product q. The right hand side shows that, in the presence of non-triangular setups, it might be better to set up from p to q and then from q to r. In the latter case, product q is considered to be a shortcut product.

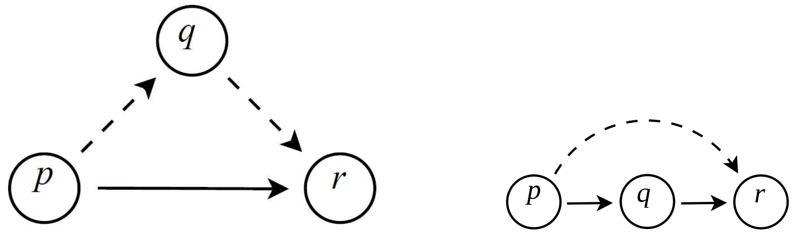

Figure 1: Triangular and Non-Triangular Setups

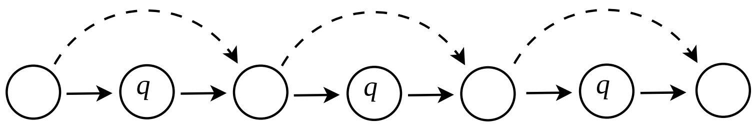

Figure 2: A Sequence of Non-Triangular Setups via a Shortcut Product q

Existing mathematical models can be used when setup times are triangular, for example, Meyr (2000) and Clark et al. (2010). Both these papers allow the production of at most one lot per product per period. However, when setup times are non-triangular then it can be optimal in certain circumstances for an intermediate shortcut product q to be produced in more than one lot within the same time-period, as shown in Figure 2. Thus the assumption of existing models (Meyr; 2000; Clark et al.; 2010) of at most one lot per product per period would not hold in such a situation. The breaking of this assumption is the key feature of the model developed below in section 4.

The GLSP models of Fleischmann and Meyr (1997) and Meyr (2000) allow non-triangular setups, as in Toso et al. (2009), but the ATSP-based model of Clark et al. (2010) assumes one lot per product per periods and so cannot allow multiple lots of shortcut products per period, as required to take advantage of non-triangular setup times. A sequence with multiple lots per period for some products could look like that illustrated in Figure 3. Subtours connected to the main sequence S by shortcut products are possible (for example, subtours B and C in Figure 3).

Thus an appropriate formulation must allow connected subtours but exclude disconnected subtours (for example, subtours A and D in Figure 3). Menezes et al. (2011) developed such a formulation using an iterative model and method based on a potentially exponential number of subtour elimination constraints. In the next section, a model is developed that uses a polynomial number of multi-commodity-flow-type constraints adapted from Claus (1984) to exclude disconnected subtours while allowing ones connected to the main sequence.

4 Modelling non-triangular setups with multiple lots per product per period

This section now develops a new model, denoted ML (*multiple lots*), for lot sizing and sequencing with asymmetric non-triangular sequence-dependent setup times. It can be viewed as a relative of the Travelling Salesman Problem with Multiple visits (TSPM) where each node is visited at least once (Punnen; 2002).

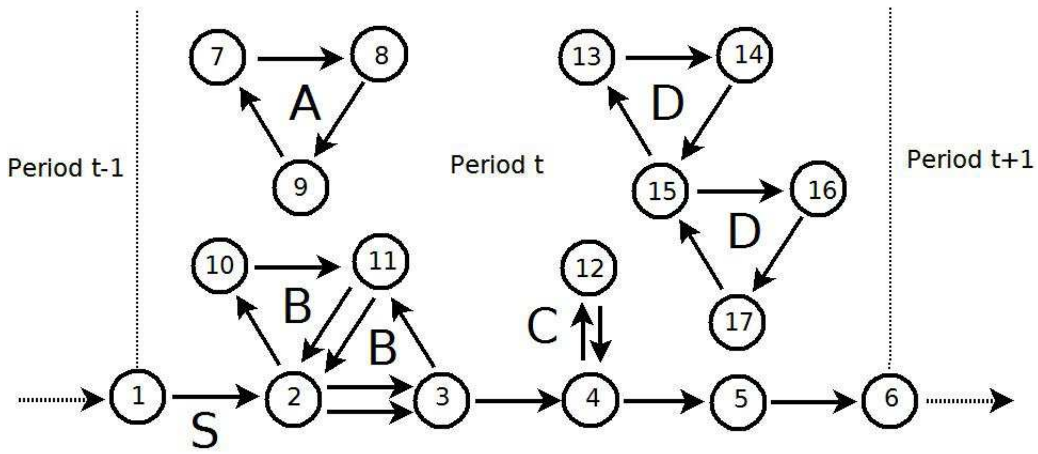

Figure 3: A main sequence (S) and different types of subtours (A,B,C,D)

Model ML considers a production process in which a set of products are to be produced over a finite planning horizon that is divided into several discrete periods. Product demands are known in advance and specified for each period. Machine capacities are taken into account in every period, as well as machine setup times which are sequence dependent, asymmetric and can be non-triangular. To prevent model infeasibility due to insufficient machine capacity, backlogs are allowed but penalized. Production of a shortcut product is allowed, even when there is no demand for it, if the objective function value is decreased by doing so. Minimum lot sizes are independent of whether the product is a shortcut or not, and are modelled by assuming that there is at least one setup in each period, as explained in section 4.5.

Production lots can cross over periods, i.e., begin in one period and finish in the next. This means that the setup state is carried over between periods. However, a product setup operation is not allowed to overlap periods, i.e., a setup begun in a given period cannot end in a subsequent period. Overcoming this limitation requires more complex modelling which will be developed in future research.

The ML model is innovative as it uses a polynomial number of constraints for prohibiting disconnected subtours which can be implemented *a priori* rather than iteratively as in Clark et al. (2010) and Menezes et al. (2011). The following indices will be used in the model description:

- p, q, r Product, $\in \{1, ..., P\}$ where P = the number of products
- t Time period, $\in \{1, ..., T\}$ where T = the number of periods (for example, days or weeks) in the planning horizon.

4.1 Input data

The input data required by the model are:

- Cap_t Available capacity time in each period t.
- u_p Time needed to produce one batch of each product p.
- ml_p Minimum lot size of product p.

h_p Inventory holding cost per period for product p.

g_p Backlog cost per period for product p.

co_t Unit cost of machine time in period t.

st_{pq} Asymmetric setup time needed to changeover from product p to product q.

d_{pt} Forecast of demand for product p at the end of period t.

I^+_{p0} Inventory of product p at the start of the planning horizon.

I^-_{p0} Backlog of product p at the start of the planning horizon.

p^α_1 The product already setup when period 1 starts, that is, the initial setup state.

4.2 Output decisions

The decisions output by the model are represented by the following variables:

I^+_{pt} Inventory of product p at the end of period t, non-negative.

I^-_{pt} Backlogs of product p at the end of period t, non-negative.

x_{pt} Total size of all the lots of product p in period t, non-negative.

$slack_t$ Number of hours of slack capacity in period t, non-negative.

y_{pqt} Number of times that production is to be changed over from product p to product q in period t, integer non-negative, $p \neq q$. For example, in Figure 3, y_{12t} = 1, and y_{23t} = 2.

z_{pt} Number of times that product p is in a setup state in period t, integer non-negative. For example, in Figure 3, z_{1t} = 1, and z_{2t} = 3.

p^α_t The product already setup when period t starts, that is, the crossover product. It is integer non-negative. Thus the model allows the setup state at the start of a period to be carried over from the previous period. Recall that p^α_1 is known, that is, it is an initial condition.

α_{pt} = 1 if p is the product already setup when period t starts (the setup state), = 0 otherwise, that is, it is binary. Note that $t \in \{1, ..., T+1\}$ and that $\alpha_{p^\alpha_t,t}$ = 1. For example, in Figure 3, α_{1t} = 1, α_{2t} = 0, and $\alpha_{6,t+1}$ = 1.

Note that the variables p^α_t and α_{pt} hold identical information. We shall use α_{pt} in the model formulation, but to allow a clear explanation we will make some use of p^α_t in the text when referring to crossover products.

4.3 Objective Function

The objective function (1) minimizes the costs of inventory and heavily penalises backlogs. It also prevents unnecessary capacity-eating setups by maximizing slack capacity (if backlogs and inventory are readily zeroed by an excess of capacity). The last term $[0.01 \sum_{p,t} z_{pt}]$ is simply a mathematical device to eliminate any excessive zero-time setups. The value of the coefficient 0.01 may need adjusting depending on the values of the other terms in (1).

$$\text{Minimise} \quad \sum_{p,t} \left(h_p \, I^+_{pt} \; + \; g_p \, I^-_{pt} \right) \; - \; \sum_t co_t \; slack_t \; + \; 0.01 \sum_{p,t} z_{pt} \tag{1}$$

4.4 Main lot size and setup constraints

Constraints (2) balance inventory, backlogs, production and demand over consecutive periods:

$$I^+_{p,t-1} \; - \; I^-_{p,t-1} \; + \; x_{pt} \; - \; d_{pt} \; = \; I^+_{pt} \; - \; I^-_{pt} \quad \forall\, p,t \tag{2}$$

The capacity constraints (3) take into account setup and production times, and calculate any capacity slack:

$$\sum_p u_p\, x_{pt} \; + \; \sum_{p,q} st_{pq}\, y_{pqt} \; + \; slack_t \; = \; Cap_t \quad \forall\, t \tag{3}$$

Constraints (4) ensure that a product can be produced in a period only if the machine is setup for it at some time in period t:

$$x_{pt} \; \leq \; M_p\, z_{pt} \quad \forall\, p,t \tag{4}$$

The coefficient M_p of z_{pt} in (4) is an upper bound on the value of x_{pt}, calculated as:

$$M_p \; = \; \min \left\{ \frac{Cap_t}{u_p}, \sum_{\tau=1}^{T} d_{p\tau} - I^+_{p0} + I^-_{p0} \right\}$$

In words, M_p is the minimum of (a) the amount of product p that can be produced if period t were entirely dedicated to its production, and (b) the effective demand for product p over all periods $t = 1, ..., T$ (given that backlogs of demand may have to be produced as well as current and future demand). The overproduction implicit in the definition of M_p is theoretically allowed, but unlikely to happen. Other externally-imposed limits on the value of x_{pt} can be incorporated into the definition of M_p in order to bring down its value so as to better fulfill its role as an upper bound on x_{pt} to enforce the "Big-M" constraints (4).

Constraints (5) prohibit setups between the same product:

$$y_{ppt} \; = \; 0 \quad \forall\, p,t \tag{5}$$

Constraints (6) ensure that there is exactly one product in a setup state at the beginning of each period:

$$\sum_p \alpha_{pt} \; = 1 \quad \text{for } t = 2, ..., T+1 \tag{6}$$

4.5 Imposing a minimum lot size

In some contexts, it may be necessary to impose a minimum lot size. In the presence of a cleansing product q, this is mandatory in order to force the proper cleaning of a previous product p's contaminants, that is, to avoid a setup from p to r via zero production of q rather than directly. Moreover, in some situations, the minimum lot size may be sequence dependent. For simplicity, this article only considers the case where the minimum lot size is not sequence dependent.

Two extra decision variables are now defined:

x^F_{pt} The minimum quantity to be produced in the first lot of product p in period t if it was setup in period t-1 (that is, p is a crossover product from period t-1 to period t), but otherwise 0 as imposed by constraints (7):

$$x^F_{pt} \; \leq \; M_p\, \alpha_{pt} \quad \forall\, p,t \tag{7}$$

x^L_{pt} The minimum quantity to be produced in the last lot of product p in period t if its production continues into period t+1 (that is, p is a crossover product from period t to period t+1), but otherwise 0 as imposed by constraints (8):

$$x^L_{pt} \;\leq\; M_p\,\alpha_{p,t+1} \quad \forall\, p,t \tag{8}$$

Constraints (10) and (11) below impose a minimum lot size on the condition that at least one product is setup in every period:

$$\sum_{p,q:p\neq q} y_{pqt} \;\geq\; 1 \quad \forall\, t \tag{9}$$

Constraints (9) are likely to exclude optimal solutions when time periods are short in length, and demand for specific products is infrequent but large when it occurs. The formulation of minimum lot-size constraints that do not require constraints (9) is a topic for future research.

When a setup state p is neither inherited from the previous period t-1 nor passed on to the next period t+1, then total production x_{pt} is composed entirely of non-crossover lots, α_{pt} = $\alpha_{p,t+1}$ = 0 and so $x^L_{pt} + x^F_{p,t+1}$ = 0 by constraints (7) and (8). In this case, constraints (10):

$$x_{pt} - x^F_{pt} - x^L_{pt} \;\geq\; ml_p\,(z_{pt} - \alpha_{pt} - \alpha_{p,t+1}) \quad \forall\, p,t \tag{10}$$

become $x_{pt} \geq ml_p z_{pt}$ so that the total x_{pt} of the lot sizes can be split into z_{pt} separate lots, each of which is at least ml_p units in size.

However, if a setup state p is either inherited from the previous period t-1 or passed on to the next period t+1 (or both), then at least some (if not all) of x_{pt} is composed of a crossover lot. In this case , either $\alpha_{pt} + \alpha_{p,t+1}$ = 1 and $z_{pt} \geq 1$, or $\alpha_{pt} + \alpha_{p,t+1}$ = 2 and $z_{pt} \geq 2$. Thus $z_{pt} - \alpha_{pt} - \alpha_{p,t+1} \geq 0$ so that constraints (10) impose $x_{pt} - x^F_{pt} - x^L_{pt} \geq 0$, that is, $x_{pt} \geq x^F_{pt} + x^L_{pt}$. Constraints (10) also then impose that the $(z_{pt} - \alpha_{pt} - \alpha_{p,t+1})$ lots of p produced entirely within period t should be of total size at least $z_{pt}\,ml_p$, again splittable into z_{pt} separate lots, each of which is at least ml_p units in size.

Constraints (11) impose a minimum size on any crossover lot:

$$x^L_{pt} + x^F_{p,t+1} \;\geq\; ml_p\,\alpha_{p,t+1} \quad \forall\, p \text{ and } t = 0, ..., T \tag{11}$$

where x^L_{p0} is known, being the amount already produced in the current lot for $p = p^\alpha_1$, the initial setup state.

Note that, if constraints (9) were not imposed, then a crossover lot that was started in period t-1 could possibly continue into t+1 and later periods. In this case, constraints (11) would become $x^L_{pt} + x^F_{p,t+1} \geq ml_p$ which is too large a lower bound for this part of the lot, given that the lot itself began earlier in period t-1.

4.6 Lot Sequencing Constraints

We have left until last the consideration of the ATSP-related constraints for sequencing product lots. Constraints (12) and (13) are flow conservation constraints that relate the α_{pt} and z_{pt} setup state variables to the y_{pqt} changeover variables, to and from a product respectively. In Figure 4, the inflow to node p is represented by the setup state variable of product p in period t (α_{pt}), and the changeover variables to p ($\sum_q y_{qpt}$). The outflow is represented by the setup state variable (α_{pt+1}) in period $t+1$ and the changeover variables from p ($\sum_q y_{pqt}$). Note that product p is included in the sequence only if there is a setup for it in period t (i.e., only if $z_{pt} = 1$).

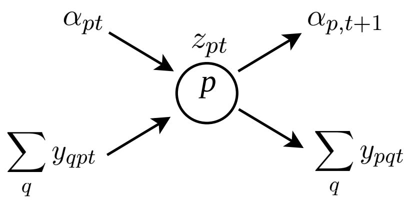

Figure 4: Node flow modelled by constraints (12) and (13)

$$\alpha_{pt} + \sum_q y_{qpt} = z_{pt} \quad \forall\, p, t \tag{12}$$

$$\sum_q y_{pqt} + \alpha_{p,t+1} = z_{pt} \quad \forall\, p, t \tag{13}$$

For example, referring to Figure 3, if p = 1, then the values in constraints (12) and (13) are 1 + 0 = 1 and 1 + 0 = 1 respectively. If p = 2, then the values are 0 + 3 = 3 and 3 + 0 = 3 respectively.

The optimal solution to the model specified by expressions (1) to (13) will consist of a single sequence starting with product $p|\{\alpha_{pt} = 1\}$ and ending with $r|\{\alpha_{r,t+1} = 1\}$ (possibly with embedded connected subtours), and maybe one or more disconnected subtours. For example, in Figure 3, the main sequence is $1 \to 2 \to 3 \to 4 \to 5 \to 6$, with 4 subtours:

A: $7 \to 8 \to 9 \to 7$

B: $2 \to 10 \to 11 \to 2 \to 3 \to 11 \to 2$

C: $4 \to 12 \to 4$

D: $13 \to 14 \to 15 \to 16 \to 17 \to 15 \to 13$

Subtours B and C are connected to the main sequence S and so permitted, but subtours A and D are disconnected and so must be prohibited. Subtour A or D can be part of a solution but only if connected to the main sequence S. How can this be modelled?

The paper by Öncan et al. (2009) reviews and analytically compares many ATSP formulations. It highlights the tightness of the *multi-commodity-flow* (MCF) formulation by Claus (1984) which is the inspiration for the formulation that prohibits disconnected subtours *a priori* in the proposed model ML. The main idea of this formulation is to ensure that, in any period t, there is always a walk from the crossover product p_t^α to any other product r in period t's sequence.

First define additional binary decision variables a_{pqt}^r as follows:

a_{pqt}^r = 1 if the arc $p \to q$ is on a walk from crossover product p_t^α to product r within period t's sequence of lots

= 0 otherwise

For any product r produced in period t, the variables a_{pqt}^r' encode a walk from p_t^α to r. It can be called an r-walk. The existence of an r-walk ensures that product r is connected to the main production sequence, maybe within a connected subtour. Figure 5 shows part of

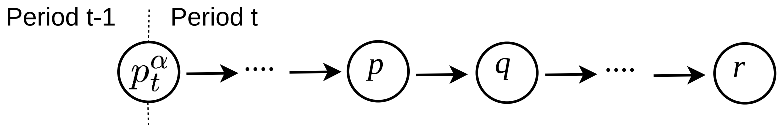

Figure 5: An r-walk from crossover product p_t^α to product r

a r-walk from the crossover product p_t^α to product r passing through the arc $p \to q$. In this case, $a_{pqt}^r = 1$.

Constraints must be formulated to enforce an r-walk for all products r produced in period t. To begin with, the arc $p \to q$ must be part of a solution in order for a_{pqt}^r to have value 1. Thus values of a_{pqt}^r must obey constraints (14):

$$a_{pqt}^r \le y_{pqt} \quad \forall\, p, q, r, t \tag{14}$$

Consider once again the infeasible sequence in Figure 3. Product r=10 in connected-subtour B is reachable from crossover product p_t^α=1 by traversing arcs $1 \to 2 \to 10$. This reachability is indicated by the following non-zero values of a_{pqt}^r that constitute an r-walk: $a_{1,2,t}^{10} = a_{2,10,t}^{10} = 1$. In contrast, product r=9 in disconnected-subtour A in Figure 3 is not reachable from crossover product p_t^α=1. No r-walk exists for r=9. This is indicated by the impossibility of finding values of a_{pqt}^9 that also obey constraints (15, 16, 18, 17, 19) below.

To prohibit disconnected subtours, further binary decision variables z_{pt}^{bin} are needed:

z_{pt}^{bin} = 1 product p is ever in a setup state in period t

= 0 otherwise

Note that $z_{pt}^{bin} = 1 \Leftrightarrow z_{pt} \ge 1$ and that $z_{pt}^{bin} = 0 \Leftrightarrow z_{pt} = 0$. For example, in Figure 3, $z_{11,t}$ = 2 so $z_{11,t}^{bin}$ = 1. This is enforced by the following constraints:

$$z_{pt} \ge z_{pt}^{bin} \quad \forall\, p, t \tag{15}$$

$$z_{pt} \le ZUB_p\, z_{pt}^{bin} \quad \forall\, p, t \tag{16}$$

where ZUB_p is a prespecified upper bound (UB) on the value of z_{pt} and must be ≥ 1. ZUB_p is automatically calculated in the computational tests below as the lesser of P (the number of products) and the size of the ordered set $\{(p,q) | st_{pq} \ge st_{pr} + st_{rq}\}$, which can be very large, but is often 1 for non-shortcut products. More detailed analysis of setup times and available production capacities might bring down the value of ZUB_p.

The three sets of constraints (17-19) explained below will now allow connected subtours, and prohibit disconnected ones *a priori*.

Firstly, constraints (17) ensure that the r-walk reaches product r (Figure 6) and is imposed only when the setup state is configured for r at least once during period t (that is, only when $z_{rt}^{bin} = 1$), but not when the setup state is never configured for r during period t, (that is, when $z_{rt}^{bin} = 0$):

$$\alpha_{rt} + \sum_p a_{prt}^r = z_{rt}^{bin} \quad \forall\, r, t \tag{17}$$

For example, the r-walk $1 \to 2 \to 10$ in Figure 3 is forced to reach product r=10 by the following instance of constraints (17):

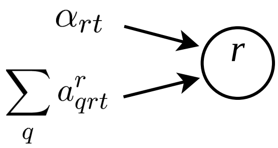

Figure 6: The r-walk from p^α_t must reach product r (if and only if $z^{bin}_{rt} = 1$)

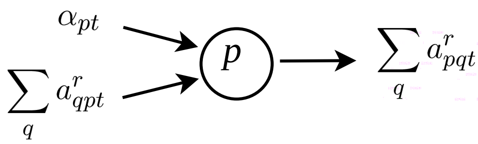

Figure 7: The r-walk from p^α_t to r may only traverse those products p for which $z^{bin}_{pt} = 1$ (and if and only if $z^{bin}_{rt} = 1$)

r = 10: $\alpha_{10,t} + \sum_p a^{10}_{p,10,t} \ = \ z^{bin}_{10,t}$ which becomes 0 + 1 = 1

and enforces that $a^{10}_{p,10,t} = 1$ for a given p.

If a product r is not produced in a period t, then $z^{bin}_{rt} \ = \ 0$, and so constraints (17) force $a^r_{prt} = 0 \ \forall \ p$ (constraints (14) also force this via $a^r_{prt} \ \leq \ y_{prt} = 0$)

Secondly, the r-walk in period t specified by the variables $\{a^r_{pqt} \mid \forall \ p, q\}$ must start at crossover product p^α_t and then traverse further products on its way to product r, as shown in Figure 7. If $\alpha_{r,t}$ = 1 then no r-walk is needed. If α_{rt} = 0, then constraints (17) mean that $\sum_p a^r_{prt} \ = \ 1$, i.e., $a^r_{prt} \ = \ 1$ for exactly one product p that is the 2nd last product on the r-walk. Constraints (18) then force $a^r_{qpt} \ = \ 1$ for exactly one product q that is the 3rd last product on the r-walk, and so on, going backwards along the r-walk, obliging the a^r_{pqt} along the r-walk to have value 1, until it reaches back to the initially-setup product $p = p^\alpha_t$ (for which α_{pt} = 1).

$$\alpha_{pt} + \sum_q a^r_{qpt} \ \geq \ \sum_q a^r_{pqt} \quad \forall \ r, p \neq r, t \tag{18}$$

For example, in Figure 3, consider the r-walk $1 \to 2 \to 10$ to product r=10. The following two instances of constraints (18) oblige the a^r_{pqt} along this r-walk to have value 1, reaching back to an initially-setup product p^α_t = 1 (for which α_{1t} is thus forced to have value 1):

p = 2: $\alpha_{2t} + \sum_q a^{10}_{q2t} \ \geq \ \sum_q a^{10}_{2qt}$ becomes $0 + \sum_q a^{10}_{q2t} \geq 1$,
resulting in $\sum_q a^{10}_{q2t} = 1$.

p = 1: $\alpha_{1t} + \sum_q a^{10}_{q1t} \ \geq \ \sum_q a^{10}_{1qt}$ becomes $\alpha_{1t} + 0 \geq 1$,
resulting in $\alpha_{1t} = 1$.

Thirdly and finally, constraints (19) require that the r-walk from p^α_t stops at product r (Figure 8) and need go no further:

$$a^r_{rqt} \ = \ 0 \quad \forall \ q, r, t \tag{19}$$

For example, the r-walk $1 \to 2 \to 10$ in Figure 3 stops at product r=10 as enforced by the following instance of constraints (19):

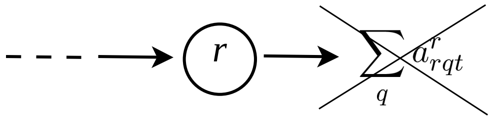

Figure 8: The r-walk from p_t^α must stop at product r (if and only if $z_{rt}^{bin} = 1$)

r = 10: $a_{10,q,t}^{10} = 0 \quad \forall\, q, t$

If r is not produced in period t, then constraints (19) simply force $a_{rqt}^r = 0 \;\forall\, p, q$, which has no impact given that constraints (17) already oblige $a_{pqt}^r = 0 \;\forall\, p, q$.

Thus constraints (17, 18, 19) exclude disconnected subtours. For example, in Figure 3, there are no instances of constraints (17) - (19) that would show that product r=9 in disconnected-subtour A is reachable by an r-walk from crossover product p_t^α. This is also true for all the other disconnected products. Thus the setup sequence in Figure 3 is infeasible and will be correctly excluded by our formulation.

4.7 Concluding the model formulation

Lastly, note that constraints (4) are valid but loose: the value of z_{pt} need only be 1, and not ≥ 2. Constraints (4) can thus be tightened by replacing z_{pt} by z_{pt}^{bin}:

$$x_{pt} \;\leq\; M_p\, z_{pt}^{bin} \quad \forall\, p, t \tag{20}$$

Thus our model, denoted ML, for lot sizing and sequencing with non-triangular setup times and setup-state carryover between periods is specified by expressions (1-3, 5-20), and restated completely in the Appendix.

Expressed as function of the number of products P and periods T, model ML has $P^2T + 7PT + T$ variables and $P^3T + 2P^2T + 11PT + 3T$ constraints, as calculated in Table 1. The ML formulation is thus polynomial-sized. This does not means that the model is solvable in polynomial time - it cannot be, given that the $\mathcal{NP}$-hard ATSP is embedded within it. Rather, the innovation in this paper has been (a) the modelling of non-triangular sequence-dependent setup times within a lot sizing model and (b) the derivation of a polynomial-sized MILP formulation for this problem.

Note that model ML is valid irrespective of whether there are non-triangular setup times or not. However, when setup times are triangular then there exists an optimal solution with zero or one lots per product per period (Clark and Clark; 2000). In this case, the formulation can then be simplified to a model that assumes *At Most One Lot* per product per period (denoted AM1L) by merging z_{pt} and z_{pt}^{bin} to be a binary variable z_{pt}. Thus constraints (15) and (16) disappear.

Model AM1L is also valid irrespective of whether the setup times are triangular or not, but in the latter case, AM1L's solution could be suboptimal given its limitation of zero or one lots per product per period. In the presence of triangular setup times, multiple lots per product per period could occur but this is not required for optimality and so in general it is avoided in models for triangular setup times. The computational tests in section 5 explore the impact of this limitation.

Variables	How many	Variables	How many	Variables	How many
I^+_{pt}	PT	I^-_{pt}	PT	$slack_t$	T
x_{pt}	PT	x^F_{pt}	PT	x^L_{pt}	PT
y_{pqt}	P^2T	z^F_{pt}	PT	α_{pt}	PT
Total number of variables = $P^2T + 7PT + T$					

Constraints	How many	Constraints	How many	Constraints	How many
(2)	PT	(3)	T	(20)	PT
(5)	PT	(6)	T	(7)	PT
(8)	PT	(11)	PT	(10)	PT
(9)	T	(12)	PT	(13)	PT
(14)	P^3T	(15)	PT	(16)	PT
(18)	$P(P-1)T$	(17)	PT	(19)	P^2T
Total number of constraints = $P^3T + 2P^2T + 11PT + 3T$					

Table 1: Number of variables and constraints in model ML

5 Computational Tests

Many models in the literature assume that there will be at most one lot per product per period. What are the pros and cons of this assumption? On the one hand, the model will be smaller with fewer variables and constraints, so we might expect faster solution times (although the tests below will show this is not so). On the other hand, the solutions with multiple lots per product per period will be excluded, so we will expect worse solutions in some cases. The computational tests in this section investigate this supposed trade-off and show that often it may not exist.

The aim of the tests was to assess how effectively the ML model took advantage of shortcut products to reduce the total time spent on setups, compared to the equivalent AM1L model. The tests also evaluated the consequences of less setup time on reducing demand backlogs (in the case of tight production capacity) or increasing the spare capacity (in the case of loose capacity), as well as the computing time of both models. The ML and AM1L models were both implemented in the AMPL modelling language (Fourer et al.; 2003) and solved using the Gurobi optimizer v4.5.0 (64-bit) (Gurobi Optimization Inc.; 2011) under Windows 7 on an Intel Core i5 CPU M460 at 2.53 GHz with 4Gb of RAM. The Gurobi optimizer was allowed to run for a maximum of 1 hour of running time, at which point the incumbent solution (i.e., the best found up to then) was used.

To obtain initial insights, the performance of both models was first compared on a system with P = 10 products whose lot sizes and sequences were to be scheduled over a horizon of T = 4 demand periods. The following data were used I_{p0} = 0.0, Cap_t = 100.0, u_p = 0.4, ml_p = 1.0, h_p = 10.0, co_t = 1.0, $\forall\ p, t$; and p^α_1 = product `P1` (arbitrarily) for all instances. The setup times were initially set to be $st_{pq} = (q - p)$ if $q > p$ otherwise $(10 + q - p)$ where $p, q \in \{1...10\}$, so that product `P2` would normally be setup immediately after `P1`. However, `P5` was then made an extreme shortcut product with zero setup times: $st_{5q} = st_{p5} = 0$. The periodic demand forecasts d_{pt} varied over product p and period t to provoke non-uniform lot-sizes and avoid lot-for-lot production. They were then randomly varied by $\pm 50\%$ within the 25 runs of each statistical experiment.

To simulate loose capacity the overall demand was adjusted so that setup times could take up to 15% of capacity, that is 15 time units per period. Tight capacity was simulated by increasing each demand d_{pt} by 20% so that setups were left with no capacity in which to occur, provoking backorders of demand.

Thus a total of 50 problem instances were generated with loose capacity. (25 with P = 10 and 25 with P = 20). Increasing each demand value by 20% then provided another 50 instances with tight capacity.

Table 2 compares the performance of both models on 6 key criteria calculated over the planning horizon:

1. Total number of setups = $\sum_{p,t} z_{pt}$
2. Total time spent on setups = $\sum_{p,q} st_{pq}\, y_{pqt}$
3. Amount of unused (slack) capacity = $\sum_t slack_t$
4. Inventory = $\sum_{p,t} I^+_{pt}$
5. Backlogs = $\sum_{p,t} I^-_{pt}$
6. CPU time = the sum of the time spend by the Gurobi optimizer and the AMPL modelling system (the latter is a few seconds at most).

For each criterion, the difference between the *mean* values for the two models were statistically tested using a balanced analysis of variance test. A similar test was carried out for the difference between the *median* values using the non-parametric Friedman test (Corder and Foreman; 2009) which is less likely to mistakenly indicate significance caused by outliers. Both tests used the data instance (that is the run) as a random blocking factor. The null hypothesis in both tests is that the difference between the model means/medians is zero.

Examining the results in Table 2, first note the highly significant increase in numbers of setups and slack capacity, and decrease in total setup time and backlogs, for the ML model compared to those for the AM1L model, particularly when capacity is tight.

For P = 10 products, model ML uses the shortcut product `P5` to economise on setups times, albeit with a larger number of actual setups, most of which (but not all) take zero time making good use of `P5`. Table 2 shows that this is particulary pronounced under tight capacity where model ML reduces the total setup time by 85%, thus keeping backlogs to a minimum. This reduction in backlogs illustrates well the economic added value of model ML over model AM1L.

Note the fast solution times for P = 10 products using the default settings of the Gurobi 4.5.0 solver. All instances of both models were solved within the maximum of 1 hour of running time

Table 2 also shows the results with twice as many products (P = 20), two extreme shortcut products (`P5` and `P15`), double the capacity per period, but T=4 still. The demand and setup times for products `P11` to `P20` simply replicate those for `P1` to `P10`. Again `P5` and `P15` were used for nearly all setups, but not always as sometimes a direct but short setup from, for example, `P4` to `P3` was more efficient as it avoided `P5` and `P15`'s minimum lot sizes.

Note the predictably much longer solution times for 20 products compared to those for 10 products. When capacity was loose, 16 of the 25 instances of the AM1L model with 20 products used the full one hour allowance of computing time (with a median optimality gap of 1.3% for these 16). This fell to 6 of the 25 instances for the ML model (with a median gap of 1.4% for these 6). When capacity was tight, 9 of the 25 instances of the AM1L model used the whole hour (with a median gap of 0.9% for these 9), while none did for the ML model.

			Mean			Median		
P	Capacity	Meas. of Perf.	AM1L	ML	p	AM1L	ML	p
10	Loose	No. of Setups	33.0	43.1	**0.000**	33.0	43.0	**0.000**
		Setup Time	22.4	11.7	**0.000**	23.0	12.0	**0.000**
		Slack Capacity	49.2	60.0	**0.000**	49.8	60.8	**0.000**
		Inventory	131.1	118.1	**0.000**	121.5	109.0	**0.000**
		Backlogs	0.00	0.00	na	0.00	0.00	na
		CPU time	13.44	7.20	**0.020**	10.0	6.0	**0.001**
P	Capacity	Meas. of Perf.	AM1L	ML	p	AM1L	ML	p
10	Tight	No. of Setups	27.1	39.0	**0.000**	27.5	39.5	**0.000**
		Setup Time	16.0	2.6	**0.000**	16.0	2.0	**0.000**
		Slack Capacity	2.94	7.96	**0.000**	0.00	2.80	**0.002**
		Inventory	268.5	302.3	**0.001**	264.2	307.2	0.162
		Backlogs	36.8	15.8	**0.000**	25.0	0.0	**0.000**
		CPU time	8.48	21.24	0.064	7.5	6.5	0.683
P	Capacity	Meas. of Perf.	AM1L	ML	p	AM1L	ML	p
20	Loose	No. of Setups	66.2	84.5	**0.000**	64.50	83.50	**0.000**
		Setup Time	17.56	2.04	**0.000**	17.5	1.5	**0.000**
		Slack Capacity	123.5	139.0	**0.000**	121.9	137.9	**0.000**
		Inventory	239.6	225.2	**0.000**	233.5	218.5	**0.000**
		Backlogs	0.00	0.00	na	0.00	0.00	na
		CPU time	3,189	1318	**0.000**	3,055	847	**0.002**
P	Capacity	Meas. of Perf.	AM1L	ML	p	AM1L	ML	p
20	Tight	No. of Setups	51.5	65.0	**0.000**	50.0	63.0	**0.000**
		Setup Time	10.5	0	**0.000**	10.00	0	**0.000**
		Slack Capacity	9.06	14.34	**0.000**	0	4.00	**0.005**
		Inventory	631.4	655.0	**0.004**	672.5	702.0	0.317
		Backlogs	33.66	20.64	**0.000**	15.0	0	**0.000**
		CPU time	2,802	187.8	**0.000**	2,771	136.0	**0.000**

Table 2: Comparison of models AM1L and ML

For both 10 and 20 products, observe that model ML solves significantly faster than AM1L, except under tight capacity for 10 products where there is not a statistically significant difference. The faster solution times of ML seem counter-intuitive given that it has more binary and integer variables than AM1L and so might be assumed to be more combinatorial complex. However, model ML does permit the obvious optimal solution in which P5 and P15 are used for nearly all setups, and so may home in more rapidly to an optimal solution than AM1L. This hypothesis needs further computational testing with other data sets.

6 Conclusions and Future Research

The theoretical contribution in this article has been the development of a new model for lot sizing and sequencing with a polynomial number of constraints that can handle the multiple lots per product per period that arise in the presence of non-triangular sequence-dependent setup times. It is a practical advantage that the model can be solved by commerciallyavailable MIP software, so that a user can readily implement the model without relying on specialist algorithms. It could still be worthwhile to develop a specialist algorithm that would accelerate the solution of the model, but this is left as a topic for future research.

The computational tests validated and confirmed that the multiple-lots feature of the model enables more efficient production than when the formulation is restricted to single lots per product per period. The model can also be faster to solve than in the latter case, despite being more complex computationally, maybe because for some problem instances (such as our tests above) there is an outstanding optimal ML solution that is quickly identified whereas an optimal AM1L solution may not be so clearly superior and hence more difficult to find.

Of the 50 ML instances with 20 products, 6 (all under Loose capacity) did not identify a provably-optimal solution within the allowed 1 hour of running time, indicating the need for future research to develop efficient solution methods for ML, possibly via exact methods such as (1) Lagrangian Relaxation coupled with decomposition into single periods where the submodels can be solved very rapidly, or via heuristic methods such as (2) *Relax-&-Fix* methods of various types (Ferreira et al.; 2009), (3) depth-first heuristics (Zhang; 2000), or (4) local branching (Fischetti and Lodi; 2003).

Future work will also computationally compare the ML model against a functionally-equivalent GLSP model (such as those based on Toso et al. (2009)'s reformulated GLSP-ST model which assumed at most one lot per product in each period) and Menezes et al. (2011)'s ATSP-based iterative method which allowed non-triangular setups.

Constraints for minimum lot sizes also need to be formulated that do not require constraints (9), that is, for crossover lots that span more than 2 periods, and also for sequence dependent minimum lot sizes.

Given that the demand forecasts usually change as time advances from one period to the next, the question arises as to whether it is worthwhile to schedule over even a medium term horizon, let alone a log-term one. Frequent rescheduling (Haase and Kimms; 1999) implies that firm schedules should really only be specified for the immediate to short term over which demand forecasts will not change (much), while approximate or aggregate planning (rather than scheduling should be carried out for medium to long term. This poses interesting (and not trivial) research challenges about how to perform planning that result in effective and efficient short term schedules (Clark; 2003).

Acknowledgements: We wish to thank the referees for their careful reading of the paper. Their comments and queries enabled us to improve its quality, clarity and readability. This research was partly supported by a *Global Research Award* from the *Royal Academy of Engineering*, London, for the first author, and an FP7 Marie Curie *International Research Staff Exchange Scheme* (IRSES) grant from the European Commission for all the authors. The third author was also partly supported by the Brazilian research agencies *Capes*, *CNPq* and *Fapesp*.

A The ML model

$$\text{Minimise} \quad \sum_{p,t} \left(h_p\, I^+_{pt} \;+\; g_p\, I^-_{pt} \right) \;-\; \sum_{t} co_t\ slack_t \;+\; 0.01 \sum_{p,t} z_{pt} \tag{1}$$

such that

$$I^+_{p,t-1} \;-\; I^-_{p,t-1} \;+\; x_{pt} \;-\; d_{pt} \;=\; I^+_{pt} \;-\; I^-_{pt} \quad \forall\, p,\; t \tag{2}$$

$$\sum_p u_p\, x_{pt} \;+\; \sum_{p,q} st_{pq}\, y_{pqt} \;+\; slack_t \;=\; Cap_t \quad \forall\, t \tag{3}$$

$$x_{pt} \;\leq\; M_p\, z_{pt}^{bin} \quad \forall\, p,\; t \qquad \text{instead of (4)} \tag{20}$$

$$y_{ppt} \;=\; 0 \quad \forall\, p,\; t \tag{5}$$

$$\sum_p \alpha_{pt} \;= 1 \quad \text{for } t = 2, ..., T+1 \tag{6}$$

$$x_{pt}^F \;\leq\; M_p\, \alpha_{pt} \quad \forall\, p,\; t \tag{7}$$

$$x_{pt}^L \;\leq\; M_p\, \alpha_{p,t+1} \quad \forall\, p,\; t \tag{8}$$

$$x_{pt}^L + x_{p,t+1}^F \;\geq\; ml_p\, \alpha_{p,t+1} \quad \forall\, p \text{ and } t = 0, ..., T \tag{11}$$

$$x_{pt} - x_{pt}^F - x_{pt}^L \;\geq\; ml_p\,(z_{pt} - \alpha_{pt} - \alpha_{p,t+1}) \quad \forall\, p,\; t \tag{10}$$

$$\sum_{p,q:p\neq q} y_{pqt} \;\geq\; 1 \quad \forall\, t \tag{9}$$

$$\alpha_{pt} \;+\; \sum_q y_{qpt} \;=\; z_{pt} \quad \forall\, p,\; t \tag{12}$$

$$\sum_q y_{pqt} \;+\; \alpha_{p,t+1} \;=\; z_{pt} \quad \forall\, p,\; t \tag{13}$$

$$a_{pqt}^r \;\leq\; y_{pqt} \quad \forall\, p,\; q,\; r,\; t \tag{14}$$

$$z_{pt} \;\geq\; z_{pt}^{bin} \quad \forall\, p,\; t \tag{15}$$

$$z_{pt} \;\leq\; ZUB_p z_{pt}^{bin} \quad \forall\, p,\; t \tag{16}$$

$$\alpha_{rt} + \sum_q a_{qrt}^r \;=\; z_{rt}^{bin} \quad \forall\, r, t \tag{17}$$

$$\alpha_{pt} + \sum_q a_{qpt}^r \;\geq\; \sum_q a_{pqt}^r \quad \forall\, r, p \neq r, t \tag{18}$$

$$a_{rqt}^r \;=\; 0 \quad \forall\, q, r, t \tag{19}$$

References

Araújo, S. A., Arenales, M. N. and Clark, A. R. (2007). Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups, *Journal of Heuristics* **13**(4): 337–358.

Carpaneto, G., Dell'Amico, M. and Toth, P. (1995). Exact solution of large scale asymmetric travelling salesman problems, *ACM Transactions on Mathematical Software* **21**(4): 394–409.

Clark, A. R. (2003). Optimization approximations for capacity constrained material requirements planning, *International Journal of Production Economics* **84**(2): 115–131.

Clark, A. R., Almada-Lobo, B. and Almeder, C. (2011). Editorial: Lot sizing and scheduling - industrial extensions and research opportunities, special issue on lot sizing and scheduling, *International Journal of Production Research* **49**(9): 2457–2461.

Clark, A. R. and Clark, S. J. (2000). Rolling-horizon lot-sizing when setup times are sequence-dependent, *International Journal of Production Research* **38**(10): 2287–2308.

Clark, A. R., Morabito, R. and Toso, E. A. V. (2010). Production setup-sequencing and lot-sizing at an animal nutrition plant through ATSP subtour elimination and patching, *Journal of Scheduling* **13**(2): 111–121.

Claus, A. (1984). A new formulation for the travelling salesman problem, *SIAM Journal on Algebraic and Discrete Methods* **5**: 21–5.

Corder, G. W. and Foreman, D. I. (2009). *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach*, Wiley-Blackwell.

Dantzig, G., Fulkerson, R. and Johnson, S. (1954). Solution of a large-scale traveling-salesman problem, *Operations Research* **2**: 393–410.

Drexl, A. and Kimms, A. (1997). Lot sizing and scheduling - survey and extensions, *European Journal of Operational Research* **99**: 221–235.

Ferreira, D., Morabito, R. and Rangel, S. (2009). Solution approaches for the soft drink integrated production lot sizing and scheduling problem, *European Journal of Operational Research* **196**: 697–706.

Fischetti, M. and Lodi, A. (2003). Local branching, *Mathematical Programming, Series B* **98**: 23–47.

Fleischmann, B. and Meyr, H. (1997). The general lotsizing and scheduling problem, *OR Spektrum* **19**(1): 11–21.

Fourer, R., Gay, D. M. and Kernighan, B. W. (2003). *AMPL - A Modeling Language for Mathematical Programming*, second edn, Duxbury Press / Brooks-Cole Publishing Company, USA. http://www.ampl.com/.

Gurobi Optimization Inc. (2011). Gurobi optimizer version 4.5.0. http://www.gurobi.com.

Haase, K. and Kimms, A. (1999). Lot sizing and scheduling with sequence dependent setup costs and times and efficient rescheduling opportunities, *International Journal of Production Economics* **66**: 159–169.

Karimi, B., Fatemi Ghomia, S. M. T. and Wilson, J. M. (2003). The capacitated lot sizing problem: a review of models and algorithms, *Omega* **31**: 365–378.

Karp, R. M. (1979). A patching algorithm for the nonsymmetric traveling-salesman problem, *SIAM Journal on Computing* **8**(4): 561–573.

Kovács, A., Brown, K. N. and Tarim, S. A. (2009). An efficient MIP model for the capacitated lot-sizing and scheduling problem with sequence-dependent setups, *International Journal of Production Economics* **118**(1): 282 – 291.

Lawler, E. L., Lenstra, J. K., Rinnoy Kan, A. H. G. and Shmoys, D. B. (1985). *The Traveling Salesman Problem - A Guided Tour of Combinatorial Optimization*, Wiley, Chichester.

Menezes, A., Clark, A. and Almada-Lobo, B. (2011). Capacitated lot-sizing and scheduling with sequence-dependent, period-overlapping and non-triangular setups, *Journal of Scheduling* **14**(2): 209–219.

Meyr, H. (2000). Simultaneous lot-sizing and scheduling by combining local search with dual optimization, *European Journal of Operational Research* **120**: 311–326.

Meyr, H. (2002). Simultaneous lotsizing and scheduling on parallel machines, *European Journal of Operational Research* **139**: 277–292.

Öncan, T., Altinel, K. and Laporte, G. (2009). A comparative analysis of several asymmetric traveling salesman problem formulations, *Computers and Operations Research* **36**(3): 637–654.

Punnen, A. P. (2002). The travelling salesman problem - application, formulations and variations, *in* G. Gutin and A. P. Punnen (eds), *The Traveling Salesman Problem and its Variations*, Kluwer.

Toso, E. A. V., Morabito, R. and Clark, A. R. (2009). Lot-sizing and sequencing optimisation at an animal-feed plant, *Computers and Industrial Engineering* **57**: 813–821.

Zhang, W. (2000). Depth-first branch-and-bound versus local search: A case study, *AAAI/IAAI Proc. 17th National Conf. on Artificial Intelligence (AAAI-2000)*, Austin, Texas, pp. 930–935. Webpage: citeseer.ist.psu.edu/470650.html.