Lot-sizing with Start-up Times *

François Vanderbeck

Engineering Department, University of Cambridge

Judge Institute, Trumpington St., Cambridge CB2 1AG, U.K.

fv@eng.cam.ac.uk http://www2.eng.cam.ac.uk/~fv

January, 1997
(revised in August 1997)

Abstract

Many practical applications of lot-sizing and scheduling problems involve start-up times. Operations research literature contains but few studies of lot-sizing models that take start-up times explicitly into account. Here, we review some of these studies, discuss the models and their complexity, and we propose further models. We consider in particular a single-stage single-mode multi-item lot-sizing model with continuous set-ups and sequence independent start-up times, which we solve using an integer programming column generation algorithm and we develop a dynamic programming procedure for the single-item subproblem that treats the initial stock as a decision variable. We also use cutting planes developed by Constantino for the multi-item polyhedra. By combining column and cut generation, the lower bounds that we obtain before branching are on average less than 2% from an optimal solution. Our algorithm solves instances with 3 to 5 items and 24 periods in an average of 50 seconds on a modern workstation, and problems with 36 periods in an average of 750 seconds. Solutions guaranteed to be within 2% of optimality are obtained in less than 75% of these times.

A manufacturer produces several products on a single production line of fixed capacity. Setting up the line to the specifications of a particular product typically requires several days that are lost to production. For each of these products, the quantities to be delivered monthly are negotiated with customers on an annual basis. The problem for the manufacturer consists in making an annual production plan that minimises production and holding costs while satisfying demand requirements. There are many applications such as this that involve making lot-sizing and scheduling decisions in the presence of

*University of Cambridge, Research Papers in Management Studies, No 22, 1996.

start-up times. Examples include the manufacturing of food products or chemicals where significant clean-ups must take place between different batches of production.

The model we consider here is a single-stage multi-item single-mode production lot-sizing and scheduling model with continuous set-ups and sequence-independent start-up times. Single mode production assumes that the machine/line can only process a single item in each time period. The machine is set-up according to the specifications of the item processed. In a so-called "continuous set-up" model, production can take any value up to capacity, once the machine is properly set up, unlike the discrete set-up model where production must be at capacity if the machine was set up. A start-up occurs when the machine is set up for an item for which is was not set up in the previous period. The associated start-up time reflects the time required to prepare the production facility, which is assumed to be independent of the production sequence (we do not use the term changeover time which is perhaps better suited for the sequence dependent case). Without the single mode assumption, it would be necessary to keep track of the actual production sequence within each time period in order to model accurately the start-ups between time periods.

For each item $i \in \{1,\ldots,n\}$, and time period $t \in \{1,\ldots,T\}$, let $d_t^i \geq 0$, $C_t^i \geq 0$, and $0 \leq L_t^i \leq C_t^i$ denote respectively the demand, the production capacity, and the lost capacity due to a start-up for item i in period t, and let $p_t^i \geq 0$, $h_t^i \geq 0$, $f_t^i \geq 0$, $g_t^i \geq 0$ denote respectively the production, the holding, the set-up and start-up cost for item i in period t. The decision variables are defined as follows:
x_t^i is the amount of item i that is produced in period t,
s_t^i is the stock level for item i at the end of period t,
y_t^i is 1 if the machine is set-up for item i during period t and is zero otherwise, and
z_t^i is 1 if a start-up occurs for item i at the outset of period t and is zero otherwise.

Then the *multi-item single-mode Lot-sizing problem with Continuous set-up and sequence independent Start-up Times* can be formulated as:

$$[LCST] \qquad \min \quad \sum_{it} p_t^i x_t^i + \sum_{it} h_t^i s_t^i + \sum_{it} f_t^i y_t^i + \sum_{it} g_t^i z_t^i \tag{1}$$

s.t.

$$\sum_i y_t^i \leq 1 \qquad \forall t \tag{2}$$

$$s_{t-1}^i + x_t^i = d_t^i + s_t^i \qquad \forall i\, t \tag{3}$$

$$x_t^i \leq C_t^i y_t^i - L_t^i z_t^i \qquad \forall i\, t \tag{4}$$

$$y_t^i - y_{t-1}^i \leq z_t^i \leq \min\{y_t^i,\, 1 - y_{t-1}^i\} \qquad \forall i\, t \tag{5}$$

$$y_t^i,\, z_t^i \in \{0,1\} \qquad \forall i\, t \tag{6}$$

$$x_t^i,\ s_t^i \ \geq \ 0 \qquad \forall i\, t \qquad (7)$$

$$s_0^i \geq 0 \quad , \quad s_T^i = 0 \qquad \forall i \qquad (8)$$

where constraints (2) enforce the single mode restriction, constraints (3) enforce the conservation of flow, constraints (4) formulate the capacity restriction including the loss of capacity due to start-up times, and constraints (5) define the start-up variables in terms of the set-up variables. As we shall see, this problem is NP-hard. Even the question of feasibility is difficult: there is no known polynomial algorithm to check feasibility a priori.We allow therefore initial stocks s_0^i to be positive, with associated costs h_0^i, which could represent the unit costs of subcontracting for instance. When costs h_0^i represent high penalties, the solution will provide information on minimal initial stock levels required to sustain a production plan over time periods $1, \ldots, T$, rather than a mere diagnostic of infeasibility. Without loss of generality, we assume that the final stocks are zero: $s_T^i = 0$ for all i.

The single item version of this problem is not only interesting in its own right but also an essential building block in a decomposition approach for solving $LCST$. The *Single-Item Lot-sizing problem with Continuous set-up and sequence independent Start-up Times* takes the form:

$$\min \qquad \sum_t p_t x_t + \sum_t h_t s_t + \sum_t f_t y_t + \sum_t g_t z_t \qquad (9)$$

[SILCST] s.t.

$$s_{t-1} + x_t \ = \ d_t + s_t \qquad \forall t \qquad (10)$$

$$x_t \ \leq \ C_t\, y_t - L_t\, z_t \qquad \forall t \qquad (11)$$

$$y_t - y_{t-1} \leq z_t \ \leq \ \min\{y_t,\ 1 - y_{t-1}\} \qquad \forall t \qquad (12)$$

$$y_t,\ z_t \ \in \ \{0, 1\} \qquad \forall t \qquad (13)$$

$$s_t\,,\ x_t \ \geq \ 0 \qquad \forall t \qquad (14)$$

$$s_0 \geq 0 \quad , \quad s_T = 0 \qquad \forall t\,. \qquad (15)$$

Note that the upper bounds on variables z_t are not necessary as $g_t \geq 0$ for all t. As we shall show, the single item problem $SILCST$ is NP-hard if either capacities or start-up times are not constant, but it is polynomially solvable when the capacities and the start-up times are constant, i.e. $C_t = C$ and $L_t = L$ for $t = 1, \ldots, T$.

In this paper, we *assume constant capacities and start-up times* for each item. Note that, assuming stationary capacities, maintenance can be accommodated in a production plan by introducing a dummy item. We present an $O(T^6)$ dynamic program for $SILCST$ that explicitly considers initial stock as a decision variable. We show how this algorithm simplifies when production and holding costs satisfy the Wagner-Whitin condition ($p_t + h_t \geq p_{t+1}\ \forall t$), yielding an $O(T^4)$ complexity. We solve $LCST$ using an IP column generation algorithm combined with a cutting plane algorithm, where the column generation subproblem takes the form of $SILCST$. The cutting planes are the inequalities that give the convex hull of feasible (y, z) solutions (Constantino, 1995). Our algorithm solves problems with 3 or 5 items and with 24 and 36 time periods in an average of 50 and 750 seconds of CPU time respectively. We begin with a review the literature on related lot-sizing models. We discuss and compare different ways of modelling start-up times, and examine the complexity of some of these models.

1 Literature

Trigeiro *et al* (1989) have studied a *Capacitated Lot-sizing model with Set-up Times* ($CLST$). Their model assumes that each item can be produced at most once in any time period. Producing a batch of item i entails a set-up time L^i and a capacity consumption B^i per unit that is produced. The resulting model includes constraints

$$\sum_i B^i\, x^i_t + L^i\, y^i_t \leq C_t \quad \forall t \tag{16}$$

in place of our single mode constraints (2) and start-up variables are absent. Trigeiro *et al* note that even the feasibility version of their model is NP-complete (by restriction to bin packing). They tackle the problem using a Lagrangian relaxation approach, dualizing constraints (16) to decompose the problem into n single-item uncapacitated lot-sizing subproblems which they solve by dynamic programming. Using sub-gradient optimization, they obtain lower bounds and dual prices which they use in a *smoothing heuristic* to construct feasible solutions from the subproblem solutions. They obtain provably near-optimal solutions for problems with up to 24 items and 30 time periods within 6 min of CPU time on an IBM 4381. Du Merle *et al* (1996) study the same model and show that the Analytic Center Cutting Plane Method (ACCPM) is computationally more efficient than the sub-gradient method in solving the Lagrangian dual.

Cattryse *et al* (1993) and van Eijl (1996) consider a *Discrete Lot-sizing model with Start-up Times* ($DLST$) which differs from $LCST$ in two ways: start-up times are modelled as integer numbers of time periods, $L^i = k^i C^i$ with $k^i \in I\!N$, during which no production can occur and production must be at capacity. Both papers assume stationary capacities and start-up times, and eliminate the production variables using the equality $x^i_t = C^i y^i_t$. The demands are normalised in terms of full production periods [1], i.e. $D^i_t = \left\lceil \frac{\sum_{\tau=1}^{t} d^i_\tau}{C^i} \right\rceil - \left\lceil \frac{\sum_{\tau=1}^{t-1} d^i_\tau}{C^i} \right\rceil$ (Magnanti and Vachani, 1990). The single mode constraints

[1]Throughout this paper, we use the notation $\lceil x \rceil$ (resp. $\lfloor x \rfloor$) to denote the smaller (resp. larger) integer not larger (resp. smaller) than x.

(2) of $LCST$ are replaced by

$$\sum_{i:k^i<t} (y_t^i + \sum_{\tau=\max\{k^i+1,t+1\}}^{\min\{t+k^i,T\}} z_\tau^i) \leq 1 \quad \forall t \tag{17}$$

since, when $z_t^i = 1$, the machine is started up for item i during the interval of periods $[t - k^i, t - 1]$. Van Eijl has shown that this formulation is tighter than that proposed by Cattryse *et al.* Salomon *et al* (1991) claim that the feasibility version of problem $DLST$ is NP-complete even when start-up times are all equal to 1, but their proof is incorrect since the proposed transformation is not polynomial.

Cattryse *et al* use a Lagrangian relaxation approach to obtain provably good solutions for $DLST$. They obtain lower bounds by solving approximately the linear programming master problem that results from the dualization of constraints (17) using a column generation procedure: the dual prices are initially obtained using a dual ascent heuristic and are refined using a few steps of a sub-gradient algorithm. The column generation subproblem is solved using a $O(k\,T\,D)$ dynamic programming algorithm where $D = \sum_t D_t$ is the total normalised demand for the item concerned. Feasible solutions are obtained using a heuristic for the set partitioning formulation of $DLST$ with the generated columns. For test problems with up to 6 items and 60 time periods, they obtain solutions with optimality gap less than 1.43% within five minutes of CPU time on an IBM-PS2 2 (Model 80 with math co-processor). Van Eijl gives facet-defining inequalities for the single item $DLST$ as well as a dynamic programming recursion (similar to that of Cattryse et al.), an associated network flow formulation, and its projection in the space of the split-variables y_{tl} and z_{tl}, where y_{tl} is one if production takes place in period t to meet the *l-th* normalised demand and z_{tl} is the associated start-up variable.

The *Discrete Lot-sizing model with Start-up Costs* ($DLSC$) but no start-up times has been the subject of several studies. Van Hoesel and Kolen (1994) give a $O(T\,D)$ dynamic programming algorithm for the single item problem and they show how this algorithm can be extended to the case of a fixed number of items n, giving rise to an $O(n\;T\;D^n)$ algorithm. When n is part of the input data, however, the problem is NP-hard even for zero start-up and production costs and constant holding costs as shown in van Eijl (1996) by restriction to 3-Partition. The feasibility problem remains easy for an arbitrary n as it amounts to verifying that cumulative demand does not exceed cumulative capacity in each period:

$$\sum_i D_{1,t}^i \leq t \quad \forall t$$

where $D_{1,t}^i$ is the cumulative normalised demand for item i up to period t. Fleischmann (1990) solves the multi-item problem using a Lagrangian based branch-and-bound procedure where the subproblems resulting from the Lagrangian relaxation of the single-mode constraints are solved by dynamic programming. Magnanti and Vachani (1990) study the polyhedra of the discrete lot-sizing model with start-up costs where set-up can be maintained even if no production takes place, i.e $x_t^i \in \{0,\ C^i\, y_t^i\}$.

The Lot-sizing problem with *Continuous set-up and Start-up Costs* (*LCSC*) but no start-up times is also a well studied model. For the single-item case, Florian and Klein (1971) have characterised extreme point solutions under concave production and inventory costs. They showed that an extreme solution can be decomposed into sub-plans, called *production sequences*, with zero initial and final inventory and at most one period for which the production level is "fractional", i.e. neither zero nor at capacity. In the case of a constant capacity, an optimal sub-plan can be computed by dynamic programming in $O(T^2)$, and therefore an optimal solution, combining sub-plans, can be obtained in $O(T^4)$. Van Hoesel and Wagelmans (1996) show that this dynamic program can in fact be implemented in $O(T^3)$ time when holding costs are linear. This improvement in complexity arises from the property that all optimal sub-plans over time interval $[k, l]$ with a fractional production period in $[k, l-1]$ can be obtained from the optimal sub-plan with fractional production period in l in linear time.

Bitran and Yanasse (1982) give properties of optimal solutions of $LCSC$ and resulting polynomial dynamic programming algorithms under various assumptions on the cost and capacity structure (non-increasing or constant costs, non-decreasing or constant capacities) and they show the NP-hardness of the model under other assumptions on the costs and capacities. Pochet and Wolsey (1993) consider a related single-item model in which the production capacity in each period can be an integer multiple of some basic capacity unit. They give a $O(T^3)$ dynamic program for the single-item problem with no capacity upper bound and a polynomial reformulation with $O(T^3)$ variables and constraints whose linear programming relaxation solves the problem. They show that when extra constraints, placing an upper bound of one basic unit on the capacities, are added, their linear programming reformulation solves Florian and Klein's model with constant capacities and linear costs. They also present facet-defining inequalities.

Finally, Constantino (1995) studies the polyhedra of various single-item and multi-item lot-sizing models, with continuous set-up under the single mode assumption, the most general of which includes *non-stationary capacities, backlogging* (i.e. negative stocks), *start-up times and lower bounds on production*; i.e. constraints (4) are replaced by

$$K_t^i\, y_t^i \le x_t^i + L_t^i\, z_t^i \le C_t^i\, y_t^i \qquad \forall\, i,\ t \tag{18}$$

where K_t^i denote a minimum batch size for item i in period t. Constantino derives families of valid inequalities (some of which consider start-up times explicitly) and associated separation routines. In particular, Constantino (1995) studies the multi-item polyhedra and shows that constraints

$$y^i_{t-1} + z^i_t \leq 1 - \sum_{j \neq i} (y^j_t - z^j_t) \qquad \forall i \text{ and } t \geq 2 \tag{19}$$

together with constraints (2) and (5), describe the convex hull of the feasible (y, z) solutions. Moreover, Constantino shows how the single-item uncapacitated model with production lower bounds (and no start-ups) can be solved in polynomial time using an extension of Florian and Klein's algorithm and he notes that the constant capacity model with start-up costs can be solved in a similar way.

Constantino conducted computational experiments with a branch-and-cut algorithm for the continuous set-up lot-sizing problem with stationary capacities, start-up times, production lower bounds, and constant costs (1995 and 1996). His results show that the Linear Programming (LP) relaxation of the "natural" formulation (of the form (1-8), where (4) is replaced by (18)) provides a weak bound with an optimality gap that is typically larger than 70%. Therefore a standard branch-and-bound procedure cannot solve instances with much more than 12 periods within reasonable time (2 hours on a modern workstation). By adding the cutting planes that he derived, Constantino solves instances with up to 5 items and 36 periods, some to optimality, others that have been interrupted after 2 hours of CPU time (on a sparc LX), with an optimality gap of no more than 7%.

2 Modelling Issues

The above review of literature presents three classes of lot-sizing models: capacitated models (such as $CLST$), discrete set-up models (such as $DLST$) and continuous set-up models (such as $LCST$). They differ mainly according to the length of their time period. The capacitated lot-sizing model is suitable when set-ups cannot carry over between periods, as is the case for example when the period length is equal to the maintenance or shutdown interval (Trigeiro *et al*, 1989). It otherwise overestimates the number of start-ups. For long maintenance interval, the assumption that at most one batch of an item can be processed within a period can be rather restrictive. Moreover, as noted by Fleischmann (1990), the underlying assumptions that demand in a period can be satisfied at any time within the period and that inventory costs only depend on the stock at the end of the period, are questionable when the time period is long.

The discrete model, with "all-or-nothing" production, is suitable in practice where production quantities are restricted to integer multiples of some minimum batch size (Salomon *et al*, 1991). In order to suit the assumptions of the discrete model, it might be necessary to consider time periods of short length and thereby a large number of periods. Indeed, consider an application with no start-up times. Let T be the "natural" number of periods in the time horizon (e.g. a three months horizon with weekly demands amounts to 13 periods). Assume that production is restricted to integer numbers of a minimum batch size and that demands and capacity C are expressed as multiples of the minimum batch size. Then, transforming the instance into an instance for the discrete model requires a discretization of time in order to match the minimum batch size. The resulting number of periods would be $T' = T\,C$, i.e. pseudo-polynomial in terms of the initial input data (unless C is fixed). If there are start-up times consisting of a fraction of the minimum batch size, time needs to be discretized even further.

When the period length does not match some minimum batch size or when the application requires a minimum production batch size but does not require the production level to be a multiple of it thereafter, the "all-or-nothing" production assumption causes an increase in stock levels and production costs. Of course, production and stock levels may be reduced in a postprocessing phase, but typically postprocessing will not entirely compensate for the extra cost resulting from the "all-or-nothing" assumption. Finally, most studies of the discrete model assume $x_t^i = C^i\,y_t^i$. A less restrictive implementation of the "all-or-nothing" production assumption allows production to be zero when the machine is set up, i.e. $x_t^i \in \{0,\ C^i\,y_t^i\}$, so as to allow for carrying out set-ups. This variant can easily be implemented in a dynamic programming algorithm.

The continuous set-up model can be viewed as a tradeoff between the two above models with a medium length time period. Here, the single-mode assumption also requires selection of a relatively short period but not as short as required to match some minimum batch size. Moreover, short start-up times do not require further discretization. The relaxation of the "all-or-nothing" production constraint reduces stock levels and allows for solutions that trade off more set-ups against savings in holding and production costs (e.g. one can produce just enough to hold on until the next period of low set-up cost). However, the assumption of model $LCST$, that start-up times are shorter than the period length, might sometimes be unsuitable, in particular if different items require start-up times of different magnitude.

Some of the drawbacks of the discrete and continuous set-up models discussed above can be overcome by modelling start-up times differently. When start-up times are shorter than the minimum batch-size, one may consider the "discrete set-up model equivalent" of $LCST$ with $L \leq C$ and $x_t = Cy_t - Lz_t$, and thereby avoid an increase in the number of periods. On the other hand, when start-up times are longer than the model period length and/or vary widely from one item to another, it might be more appropriate to model start-up times as $L = k\,C + \sigma$, with $k \in I\!N$ and $0 \leq \sigma < C$. The model with $L = k\,C$ may be viewed as an approximation of this model, which can be justified by lower computational complexity.

In Table 1, we summarize the different ways of modelling start-up times for discrete and continuous set-up single-item lot-sizing models. We also show the complexity of a dynamic programming solution of the problem with stationary capacities and start-ups, when known. For the capacitated model, a start-up time $L \leq C$ is probably the only option, given that the period is rather long. We emphasize that, even if start-ups are modelled in the same way, the discrete model might require a finer discretization of time than the continuous set-up model in order to overcome the drawbacks of "all-or-nothing" production assumption. On the other hand, for a fixed number of periods, the discrete model is typically easier to solve than the continuous set-up model.

MODEL	Discrete Set-up	Continuous Set-up
$L \leq C$	$x_t = C\,y_t - L\,z_t$ $O(T^3)$	$x_t \leq C\,y_t - L\,z_t$ $O(T^6)$
$L = k\,C$	$x_t = C\,y_t$ $y_t + \sum_{\tau=t+1}^{t+k} z_\tau \leq 1$ $O(k\,T\,D)$	$x_t \leq C\,y_t$ $y_t + \sum_{\tau=t+1}^{t+k} z_\tau \leq 1$
$L = k\,C + \sigma$	$x_t = C\,y_t - \sigma\,z_t$ $y_t + \sum_{\tau=t+1}^{t+k} z_\tau \leq 1$	$x_t \leq C\,y_t - \sigma\,z_t$ $y_t + \sum_{\tau=t+1}^{t+k} z_\tau \leq 1$

Table 1: Single-item single-mode lot-sizing models with constant capacity and start-up times.

In the sequel, we shall consider the model with start-up times $L \leq C$, as this is the model that suits the application that motivates this study. We leave the more general case where $L = k\,C + \sigma$ to further research which would presumably combine the results presented here with those of the discrete model where $L = k\,C$.

3 Complexity Issues

Let us now examine the complexity of the capacitated model, the continuous set-up model, and the discrete set-up lot-sizing model, under the assumption that $L \leq C$. For all three models, the multi-item problem is NP-hard. However, their complexity differs when the number of items is fixed. The complexities of the associated feasibility problems and that of the single-item subproblems that arise from decomposition also differ. For the capacitated model, the relaxation of the capacity constraints (16) gives rise to the Wagner-Whitin uncapacitated lot-sizing problem, solvable by dynamic programming in $O(T \log T)$ — Wagelmans, van Hoesel and Kolen (1990), Federgruen and Tzur (1991), and Aggarwal and Park (1993).

For the continuous and discrete set-up model, the subproblem resulting from the dualization of the single mode constraints (2) is NP-hard when capacities or start-up times are non-stationary. First, consider the case of the continuous set-up model. Florian *et al* (1980) have proved that the single-item continuous set-up lot-sizing model with constant demands, linear production costs, zero holding costs, unit set-up costs and no start-ups is NP-hard when capacities are not constant (by restriction to SUBSETSUM). We can easily show that the problem with stationary capacities but non-stationary start-up times $L_t \leq C$ is also NP-hard.

Proposition 1 *Problem SILCST is NP-hard, even for constant capacities, zero holding costs and zero start-up costs.*

The result can be established by showing that an instance of the single-item continuous set-up lot-sizing model with no start-ups and non-stationary capacities C_t for $t = 1, \ldots, T$ can be polynomially transformed into an instance of the problem with constant capacity but non-stationary start-up times. Let $C = \max_t C_t$ and $L_t = C - C_t$ for $t = 1, \ldots, T$. Then, replace each period t by two successive periods t and t'. Each newly-defined period t' has zero demand, $d_{t'} = 0$, and a large set-up cost, $f_{t'} = M$, where M is sufficiently large to make any solution that includes a set-up in a period t' suboptimal. Then, in an optimal solution to the newly-defined problem, a start-up occurs in every period t where production takes place and therefore the production level is bounded by $C_t = C - L_t$.

In the discrete set-up lot-sizing with no start-up times and non-stationary capacities, production levels are defined as $x_t = C_t y_t$. This problem is NP-hard, as it is easily seen to encompass the SUBSETSUM problem as a special case: let the capacities be $0 < C_t < C$ for $t = 1, \ldots, T-1$ and $C_T = C$, let $d_T = C$ and $p_T = 1$, all other costs and demands being zero; then there exists a subset $S \subseteq \{1, \ldots, T-1\}$ such that $\sum_{t \in S} C_t = C$ if and only if the discrete lot-sizing solution has zero cost. Hence we have proved the following:

Proposition 2 *When capacities are not constant, the discrete set-up single-item lot-sizing problem is NP-hard, even for zero holding costs, zero set-up costs, zero start-up costs, and zero start-up times.*

When capacities are constant but there are non-stationary start-up times, the problem is also NP-hard. The argument is the same as for the continuous set-up model: one can force start-up in each production period by introducing an artificial period with high set-up cost between each pair of periods. Hence,

Corollary 3 *The single-item discrete set-up lot-sizing problem with constant capacities C but non-stationary start-up times $L_t \leq C$ is NP-hard, even for zero holding costs and zero start-up costs.*

Having shown the NP-hardness of single-item problems when capacities or start-up times are not constant, we shall assume stationary capacities and start-up times in the rest of this paper. Under this assumption, the single-item lot-sizing model with start-up times is polynomially solvable by dynamic programming both for discrete and continuous set-up models as shown in the next section. Moreover, for the discrete set-up model, the dynamic programming algorithm can be extended to the case of a fixed number of items. We conjecture that the continuous set-up model is NP-hard even for a fixed number of items. For the multi-item capacitated model $CLST$, Bitran and Yanasse (1982) proved that the problem with no set-up times, constant set-up costs, zero holding cost, non-increasing production costs, and constant capacities is NP-hard if the number of items is 2 or more and that the problem with 3 items or more is NP-hard even when production costs are zero but holding costs are constant.

When the number of items is not fixed but is part of the data, the discrete set-up model is NP-hard even when start-up times are not present (van Eijl, 1996). Hence the continuous set-up model is also NP-hard because:

Observation 4 *The optimization problem for a lot-sizing model with continuous set-up and no start-up times is at least as hard as that for the associated discrete set-up model.*

Indeed, given a problem with discrete set-ups, one can normalise the demands by letting $D^i_{1,t} = \lceil \frac{d^i_{1,t}}{C} \rceil$, redefining capacities as $C^i = 1$, for $i = 1, \ldots, n$ to solve the problem with an algorithm designed for the associated continuous model. The solution will not contain any "fractional production" and will thus be feasible for the discrete set-up model. Note that the same observation holds where start-up times suit the model $L = kC$.

Let us now consider the feasibility problems associated with all three multi-item lot-sizing models with start-up times, when initial stocks are not decision variables. We first observe that any feasible solution to the discrete set-up model is also feasible for the continuous set-up model and, inversely, a feasible solution to the continuous set-up model can be modified by raising production levels to capacity in order to give a feasible solution to the discrete set-up model. Therefore,

Observation 5 *Under the single-mode constraints, the continuous set-up lot-sizing problem with start-up times is feasible if and only if the associated discrete set-up problem is feasible.*

OPTIMISATION	For a fixed number of items n	When n is not fixed
capacitated model	NP-hard (even if $L = 0$) Bitran and Yanasse (1982)	NP-hard (even if $L = 0$)
continuous set-up model	open	NP-hard (even if $L = 0$)
discrete set-up model	polyn. solvable	NP-hard (even if $L = 0$) (van Eijl, 1996)

Table 2: Complexity of multi-item lot-sizing optimization problems with stationary capacities C^i and start-up times $L^i \leq C^i$.

Since the discrete set-up optimization problem with start-up times is polynomially solvable when the number of items is fixed, the same is true of the feasibility problem. When the number of items is not fixed, Salomon et al. (1991) conjecture that the feasibility problem of the discrete model with $L = C$ is NP-complete. For the capacitated model, Trigeiro *et al* (1989) show the NP-completeness of the feasibility problem, but the proposed restriction does not apply for a fixed number of items. In Tables 2 and 3, we summarize the complexity results for the optimization and feasibility problems respectively.

FEASIBILITY	For a fixed number of items n	When n is not fixed
capacitated model	open	NPC Trigeiro *et al* (1989)
contin. and discr. set-up models	polyn. solvable	open

Table 3: Complexity of multi-item lot-sizing feasibility problems with stationary capacities C^i and start-up times $L^i \leq C^i$.

4 Solution Approach

A standard solution approach to multi-item lot-sizing problems consists in using Lagrangian relaxation to decompose the problem into single-item subproblems. In the case of $LCST$, dualizing the single mode production constraints (2) with associated weights π_t, decomposes the problem into single-item subproblems of the form:

$$\text{[SP]} \quad \begin{aligned} \min \quad & h_0\, s_0 + \sum_t c_t x_t + \sum_t e_t y_t + \sum_t g_t z_t && && (20) \\ \text{s.t.} \quad & && && \\ s_0 + x_{1,t} \;\geq\; & d_{1,t} && \forall t && (21) \\ s_0 + x_{1,T} \;=\; & d_{1,T} && && (22) \\ x_t \;\leq\; & C\, y_t - L\, z_t && \forall t && (23) \\ z_t \;\geq\; & y_t - y_{t-1} && \forall t && (24) \\ y_t\,,\; z_t \;\in\; & \{0,1\} && \forall t && (25) \\ s_0\,,\; x_t \;\geq\; & 0 && \forall t && (26) \end{aligned}$$

This formulation SP of the single item subproblem has been obtained from formulation $SILCST$ (9) by eliminating the stock variables $s_1, \ldots, s_T$ (using equations (3)) and by defining cumulative productions $x_{1,t} = \sum_{\tau=1}^{t} x_\tau$ and cumulative demands $d_{1,t} = \sum_{\tau=1}^{t} d_\tau$. The modified production costs are defined as $c_t = p_t + h_{t,T}$ where $h_{t,T}$ are the cumulative holding costs. The cost of the initial inventory has also been modified to $h_0 = h_0^{SILCST} + h_{1,T}$. The constant term $-\sum_{t=1}^{T} h_t\, d_{1,t}$ is not included in the objective function of formulation SP. The set-up costs are defined as $e_t = f_t + \pi_t$. Moreover, we assume that the capacities and start-up times are constant.

Problem $LCST$ can be reformulated in terms of the feasible solutions to these subproblems. Let $q \in Q^i = \{(x,y,z,s) \in I\!N^T \times \{0,1\}^T \times \{0,1\}^T \times I\!N^{T+1} : s_{t-1} + x_{1,t} = d_t^i + s_t\,,\; x_t \leq C^i\, y_t - L^i\, z_t\,,\; z_t \geq y_t - y_{t-1}$ for all $t\,,\; s_0 \geq 0\,,\; s_T = 0\, \}$ denote a feasible production planning for item i, $c_q^i = \sum_t p_t^i x_t + \sum_t h_t^i s_t + \sum_t f_t^i y_t + \sum_t g_t^i z_t$ be the associated cost, $y^q \in \{0,1\}^T$ be the associated set-up vector, and λ_q^i be one if production plan q is selected for item i in the solution and zero otherwise. Then, an alternative formulation for $LCST$, the so-called *master* formulation, is

$$\text{[M]} \quad \begin{aligned} \min \quad & \sum_i \sum_{q \in Q^i} c_q^i \lambda_q^i && && (27) \\ \text{s.t.} \quad & && && \\ \sum_i \sum_{q \in Q^i} y_t^q \lambda_q^i \;\leq\; & 1 && \forall t && (28) \\ \sum_{q \in Q^i} \lambda_q^i \;\geq\; & 1 && \forall i && (29) \\ \lambda_q^i \;\in\; & \{0,1\} && \forall\, i\; q \in Q^i && (30) \end{aligned}$$

where constraints (28) mean that the machine can process at most one item in each period and constraints (29) ensure that a production plan for each item will be chosen. It is well known that the linear programming relaxation of the master formulation M is equal to the Lagrangian dual obtained by taking the maximum over all $\pi \geq 0$ of the Lagrangian relaxation described above. It provides a lower bound on the integer solution typically tighter than that of the LP relaxation of $LCST$.

We solve the master problem M using an integer programming column generation algorithm (Vanderbeck and Wolsey, 1996), also known as a branch-and-price algorithm. The algorithm is a branch-and-bound procedure, where the bounds are obtained by solving the linear programming relaxation of the master, using a standard column generation procedure. At each iteration, n column generation subproblems of the form SP (20-26) are solved in search for columns q of negative reduced costs $\overline{c}^i_q = c^i_q + \sum_t \pi_t \, y^q_t - \sigma^i$, where π and σ are the dual variables associated with constraints (28) and (29) respectively. These subproblems are solved by dynamic programming as detailed below. Because there is a one-to-one correspondence between the master LP solution and that of the LP relaxation of $LCST$, branching can be implemented in a straightforward manner by fixing fractional set-up variables to their integer value (Vanderbeck, 1995). We branch on the fractional set-up variable with lowest time period index and we break ties by taking the variable which is most fractional (closest to $\frac{1}{2}$).

We use cutting planes to tighten the master LP formulation at each node of the branch-and-bound tree. Indeed, Constantino (1995) has shown that constraints (19), together with constraints (2) and (5), describe the convex hull of the feasible (y, z) solutions to $LCST$. Inequalities (19) can be expressed in terms of the master variables as

$$\sum_{q \in Q^i} (y^q_{t-1} - y^q_t + 2\, z^q_t)\, \lambda^i_q \; + \; \sum_j \sum_{q \in Q^j} (y^q_t - z^q_t) \lambda^j_q \leq 1 \quad \forall i \text{ and } t \geq 2 \qquad (31)$$

where z^q is the start-up vector in production plan q. The separation routine for these cuts is a simple enumeration procedure that runs in $O(n\, m\, T)$ time, where m is the number of non-zero master variables in the current solution which is bounded above by the number of master constraints, initially equal to $n + T$. After completion of the column generation procedure, we search for violated cuts of this form. If one is found, it is added to the master and the column generation procedure is recalled. In this way, the LP bound we obtained is that over the intersection of the convex hull of the feasible (y, z) solutions defined by constraints (2) and (5-6) and the convex hulls of the single item feasible solutions defined by (10-15).

4.1 Dynamic Program for the discrete set-up single-item problem

The discrete set-up single-item lot-sizing problem with stationary start-up time and capacity, $0 \leq L < C$, constitutes a building block for the solution of the associated continuous set-up problem. In particular, we use solutions to the discrete problem for the initialisation of the dynamic programming solution of the continuous problem. A solution to the discrete problem can also be used as a heuristic solution for the continuous model after it has been processed to reduce stock levels. Moreover, as we shall also see, a solution of the continuous model can be viewed as intervals of "all-or-nothing" productions separated by "fractional production periods" where $0 < x_t < Cy_t - Lz_t$.

Consider the discrete problem defined by (20, 21, 24-26) and the constraints that either $x_t = 0$ or $x_t = C\,y_t - L\,z_t$ $\forall t$, and $0 \leq s_T < C$. Observe that in an optimal solution to this discrete problem with non-negative costs, there exists a period $p \in [0, T]$ with $s_p = 0$ (otherwise the initial stock can be reduced by an amount equal to the minimum stock level) and hence the solution can be decomposed into two production sub-plans: one over periods 1 to p and one over period $p+1$ to T if $p < T$. Let us refer to the period p of zero inventory as the *pivot* period. We can then compute an optimal production plan over time interval $[1, p]$, where an initial inventory $s_0 \geq 0$ is allowed, using a backward dynamic programming recursion. Let $F(y_p, p)$ be the optimal cost of a feasible production plan over interval $[1, p]$ with final set-up state equal to y_p. Similarly, an optimal production plan over interval $[p+1, T]$, where a final inventory is allowed, can be obtained using a forward dynamic programming recursion. Let $G(y_p, p+1)$ be the optimal cost of a feasible production plan over time interval $[p+1, T]$ given that the set-up state in period p was y_p.

First we show how to compute $F(y_p, p)$. For a **fixed pivot** p **and final set-up state** y_p, let $U^t(y, u, q)$ be the minimum cost for a production plan over periods $t, \ldots, p$ that includes exactly u start-up periods with production level $L - C$ and exactly q full production periods with production level C, given that the set-up variable in period $t-1$ takes value y, i.e. $y_{t-1} = y$. This cost is only defined if $u(C-L) + q\,C \leq d_{t,p}$, for otherwise there would be some backlog, and if $2\,u - y + q \leq p - t + 1$, i.e. if the interval $[t, p]$ is large enough to accommodate u start-up periods and q full production periods. Initially,

$$U^{p+1}(y_p, 0, 0) = 0 \text{ and } U^{p+1}(1 - y_p, 0, 0) = \infty \ .$$

Then, for $t = p$ down to 1 and for all feasible states (y, u, q), $U^t(y, u, q)$ can be computed recursively as follows:

$$U^t(0,u,q) = \min \begin{cases} g_t + e_t + (C-L)\,c_t + U^{t+1}(1,u-1,q) \\ g_t + e_t + U^{t+1}(1,u,q) \\ U^{t+1}(0,u,q) \end{cases}$$

$$U^t(1,u,q) = \min \begin{cases} e_t + C\,c_t + U^{t+1}(1,u,q-1) \\ e_t + U^{t+1}(1,u,q) \\ U^{t+1}(0,u,q) \end{cases}$$

where the three terms in the minimisation correspond respectively to producing, setting up the machine without producing, and setting the machine off. The costs corresponding to infeasible states are implicitly assumed to be infinity. Then the optimal cost of a production plan over interval $[1,p]$ is

$$F(y_p,p) = \min_{(u,q)} \{\, U^1(0,u,q) + (d_{1,p} - u\,(C-L) - q\,C)\,h_0 \} \tag{32}$$

where the only pairs (u,q) for which $U^1(0,u,q)$ is defined are those for which $2\,u-1+q \le p$, $u\,(C-L)+q\,C \le d_{1,p}$, and $q>0 \Rightarrow u>0$ since $y_0=0$. These computations take $O(p^3)$ time.

The forward recursion that yields the value $G(p+1,y_p)$ is similar. Then the minimum cost of a solution to the discrete set-up problem can be obtained as

$$\min_{p=0,\ldots,T\,,\; y_p\in\{0,1\}} \{F(y_p,p) + G(y_p,p+1)\}$$

in $O(T^4)$. Observe, however, that when the initial stock is not a decision variable, as is the case for the standard single-item discrete lot-sizing problem, the optimal cost is $G(0,1)$ and is computed in $O(T^3)$ time. The multi-item discrete lot-sizing problem with no initial stocks (resp. no final stocks) can be solved in polynomial time when the number of items, n, is fixed using an extension of the forward (resp. backward) dynamic program for the single item problem: then, in each period t, the possible states are defined by a triplet $(i,\vec{u},\vec{q})$ where i is the item being processed in period t ($i=0$ if the machine is not set-up), $\vec{u}$ and $\vec{q} \in I\!N^n$ are vectors whose component indicate respectively the current number of start-up production periods and the current number of full production periods for each item.

4.2 Dynamic Program for the continuous set-up single-item problem

For the single item continuous set-up lot-sizing model with start-up cost, Florian and Klein (1971) showed that extreme solutions have the property that the inventory goes to zero between any two "fractional" productions, i.e. if $0 < x_k < C$ and $0 < x_l < C$, then there is a period $t \in [k, l-1]$ such that $s_t = 0$ (see also Pochet and Wolsey, 1993). This property carries over to the problem with start-up times, $SILCST$. The argument is the same. For fixed values of y and z, the problem in variables x and s is a network flow problem. In an extreme solution (x, s), the basic arcs, i.e. the arcs for which the flow is neither at its lower nor upper bound, do not form any cycle, which is equivalent to saying that the inventory goes to zero between any two *fractional* productions $0 < x_t < Cy_t - Lz_t$. In particular, if the initial inventory is not zero, the inventory must go to zero before the first fractional production period, since the initial inventory can be viewed as a fractional production in period 0. We have shown therefore that

Proposition 6 *Extreme solutions of $SILCST$ can be decomposed into sub-plans separated by periods of zero inventory, called regeneration point, and containing at most one fractional production.*

The decomposition is not unique and allows for different characterisations of "valid sub-plans" or *regeneration intervals*. In any case, an optimal solution to $SILCST$ that combines regeneration intervals can be obtained by a shortest path or a dynamic programming algorithm in $O(T^2)$ time, once the optimal costs of regeneration intervals have been computed. When capacities and start-up times are constant, as we assumed in SP, the optimal sub-plans can them-selves be computed in polynomial time.

Here, a *regeneration interval* is defined as a set of consecutive periods $[k, \ldots, l]$ such that $s_{k-1} = s_l = 0$ (i.e. $s_0 + x_{1,t} = d_{1,t}$ for $t = k-1$ and $t = l$), $s_t > 0$ for all $t \in \{k, \ldots, l-1\}$ such that $d_{kt} > 0$, and there is at most one *fractional production period* $f \in \{k, \ldots, l\}$ such that $0 < x_f < Cy_f - Lz_f$ while for all $t \in \{k, \ldots, l\} \setminus \{f\}$, either $x_t = 0$, or $x_t = Cy_t - Lz_t$. The strict positiveness of intermediate stock levels is not necessary but limits the number of regeneration intervals being considered. Let $G(y_{k-1}, y_l, k, l)$ be the minimum cost of a regeneration interval $[k, \ldots, l]$ given that the set-up variables in period $k-1$ and l are fixed to y_{k-1} and y_l respectively. Then, an optimal solution to SP can be obtained from the $G(y_{k-1}, y_l, k, l)$ values. Let $F(y_t, t)$ be the minimum cost of a feasible production plan over periods 1 to t if the set-up variable in period t takes value y_t. Initially,

$$F(0,0) = 0 \quad \text{and} \quad F(1,0) = \infty \, .$$

Then, for $t = 1, \ldots, T$, $F(y_t, t)$ is updated as follows:

$$F(y_t, t) = \min\{\tilde{F}(y_t, t) \, , \, \min_{0<k\leq t} \{F(0, k-1) + G(0, y_t, k, t) \, , \, F(1, k-1) + G(1, y_t, k, t)\}\}$$

where $\tilde{F}(y_t, t)$ represents the minimum cost of a regeneration interval $[1, t]$ with fractional production in period zero, i.e. with initial inventory, it is obtained by solving the associated discrete set-up problem and is defined by (32). The optimal solution to SP is

$\min\{F(0,T), F(1,T)\}$.

We now turn to the computation of the minimum cost of a regeneration interval k,l $G(y_{k-1}, y_l, k, l)$'s. If there are exactly s *start-up production*, i.e. production periods where $x = (C-L)$, during the regeneration interval $[k,\ldots,l]$, then there must be

$$p = \left\lfloor \frac{d_{kl} - s\,(C-L)}{C} \right\rfloor$$

full capacity production periods (where $x = C$) and the amount produced during the *fractional production* period is

$$r = d_{kl} - s\,(C-L) - p\,C\ .$$

For **fixed values** k, l, s, y_{k-1}, and y_l, let $V^t(y_t, \phi, u, q)$ denote the minimum cost of a feasible production sequence over the time interval k,t that includes u start-up periods with production at capacity $C-L$, q periods with production at full capacity C, and $\phi \in \{0,1\}$ fractional production period with production at level $r < C$, given that the values of the set-up variables in period t is y_t. Then,

$$G(y_{k-1}, y_l, k, l) = \min_s\ V^l(y_l, \delta(r), s, p),$$

where $\delta(x) = 1$ if $x > 0$ and zero otherwise.

Feasibility considerations restrict the number of states (y_t, ϕ, u, q) and possible number of start-ups s to be considered. Let ψ^t be the set of feasible states (y_t, ϕ, u, q) in period t: for $t = k, \ldots, l-1$,

$$\psi^t = \{(y_t, \phi, u, q) \in \{0,1\} \times \{0,1\} \times \mathbb{N} \times \mathbb{N} :$$

$$\phi \leq \delta(r)\ ;\ u \leq s\ ;\ q \leq p$$

$$y_{k-1} + u + \phi\,\delta(C - L - r + 1) \geq \delta(d_{kt}) \tag{33}$$

$$2\,u - 1 + y_{k-1} + q + \phi \leq t - k + y_t \tag{34}$$

$$2\,(s-u) - 1 + y_t + (p-q) + (\delta(r) - \phi) \leq l - t - 1 + y_l \tag{35}$$

$$u\,(C-L) + q\,C + \phi\,r \geq d_{kt} + \delta(d_{kt})\ \} \tag{36}$$

and

$$\psi^l = \{(y_l, \delta(r), s, p)\}\ ,$$

where (33) enforces the condition that there must be a start-up before the first positive demand if $y_{k-1} = 0$, (34) specifies that the number of periods required to accommodate u start-ups, q full production, and ϕ fractional periods cannot exceed the number of available periods up to t, similarly (35) enforces a lower bound on the number of busy periods that must have been accommodated by time t if one is to meet the target $(s, p, \delta(r))$ by the end of the interval, (36) enforces strictly positive intermediate stock levels. Lower and upper bounds on s can be derived from constraints (33) and (34) respectively:

$$s \geq \delta(\left\lceil \frac{d_{kl}}{C} \right\rceil - \left\lfloor \frac{d_{kl}}{C} \right\rfloor) - y_{k-1}$$

$$s \leq \min\{ \left\lfloor \frac{(l-k+y_l+1-y_{k-1})\, C - d_{kl}}{C+L} \right\rfloor , \left\lfloor \frac{d_{kl}}{C-L} \right\rfloor \} .$$

In the sequel, it is implicit that $V^t(y_t, \phi, u, q) = \infty$ if $(y_t, \phi, u, q) \notin \psi^t$.

The minimum costs $V^t(y_t, \phi, u, q)$ are computed recursively. Initially,

$$V^{k-1}(0,0,0,0) = \begin{cases} 0 \text{ if } y_{k-1} = 0 \\ \infty \text{ otherwise} \end{cases}$$

$$V^{k-1}(1,0,0,0) = \begin{cases} 0 \text{ if } y_{k-1} = 1 \\ \infty \text{ otherwise} \end{cases}$$

Then, for all $t = k, \ldots, l$, and for all u and q such that $(y_t, \phi, u, q) \in \psi^t$,

$$V^t(0,0,u,q) = \min \begin{cases} V^{t-1}(0,0,u,q) \\ V^{t-1}(1,0,u,q) \end{cases}$$

$$V^t(0,1,u,q) = \min \begin{cases} V^{t-1}(0,1,u,q) \\ V^{t-1}(1,1,u,q) \end{cases}$$

$$V^t(1,0,u,q) = \min \begin{cases} V^{t-1}(0,0,u,q) + g_t + e_t \\ V^{t-1}(0,0,u-1,q) + g_t + e_t + (C-L)c_t \\ V^{t-1}(1,0,u,q) + e_t \\ V^{t-1}(1,0,u,q-1) + e_t + Cc_t \end{cases}$$

$$V^t(1,1,u,q) = \min \begin{cases} V^{t-1}(0,0,u,q) + g_t + e_t + r\, c_t & \text{if } r \leq C - L \\ V^{t-1}(0,1,u,q) + g_t + e_t \\ V^{t-1}(0,1,u-1,q) + g_t + e_t + (C-L)c_t \\ V^{t-1}(1,0,u,q) + e_t + r\, c_t \\ V^{t-1}(1,1,u,q) + e_t \\ V^{t-1}(1,1,u,q-1) + e_t + Cc_t \end{cases}$$

In this straightforward implementation, $V^l(y_l, \delta(r), s, p)$ is obtained in $O(T^3)$, for any fixed values k, l, s, y_{k-1}, and y_l. Therefore, the overall complexity of the dynamic programming algorithm is $O(T^6)$. In light of the work by Van Hoesel and Wagelmans (1996), it might be possible to improve upon this complexity.

4.3 Dynamic Program for the continuous set-up single-item problem with Wagner-Whitin costs

Subproblem $SILCST$ is easier to solve when production and holding costs satisfy the so-called Wagner-Whitin condition, i.e. when $p_t + h_t \geq p_{t+1}$ $\forall t = 1, \ldots, T-1$. In particular, this condition is satisfied when production costs are non-increasing over time ($p_t \geq p_{t+1}$) or constant and holding costs are non-negative ($h_t \geq 0$), as is the case in many applications. Then, in formulation SP, the modified production costs c_t are non-increasing and it is optimal to produce as late as possible among the periods in which a set-up is performed. Thus, in a regeneration interval, the fractional production period, if any, will be the first period in which production occurs. In the words of Bitran and Yanasse (1982), there exists an optimal solution with the property that

$$s_{t-1}\, x_t\, (C y_t - L z_t - x_t) = 0 \;\; \forall t$$

and $0 < x_t < (C y_t - L z_t) \Rightarrow s_{t-1} = 0$. This characterisation of optimal solutions reduces the complexity of the dynamic program above to $O(T^4)$, since computation of optimal discrete set-up production sub-plans can be done jointly for all regeneration intervals $[k, l]$ sharing the same end point l.

We now briefly present a backward dynamic program that solves problem SP with Wagner-Whitin costs. Let $U^t(y, k, u, q)$ be the minimum cost for a feasible production plan over periods $t, \ldots, T$, where the first regeneration point is $k \in [t, T+1]$, (i.e. $k = \text{argmin}\{\tau \geq t : s_{\tau-1} = 0\}$) and there are exactly u start-up periods with production at level $(L - C)$ and q full production periods with production at level C in the interval $[t, k-1]$, given that the set-up variable in period $t-1$ takes value y, i.e. $y_{t-1} = y$. When $k = t$, this cost is only defined for $u = q = 0$. When $k > t$, this cost is defined for pairs (u, q) such that $u\,(C - L) + q\, C < d_{t,k-1}$ (insuring a positive stock $s_t > 0$) and $2\,u - y + q \leq k - t$ (the interval $[t, k-1]$ must be large enough to accommodate u start-up periods and q full production periods). Initially,

$$U^{T+1}(0, T+1, 0, 0) = U^{T+1}(1, T+1, 0, 0) = 0$$

Then, for $t = T$ down to 1 and for all feasible states (y, k, u, q) with $k > t$, $U^t(y, k, u, q)$ can be computed recursively as follows:

$$U^t(0, k, u, q) = \min \begin{cases} g_t + e_t + (C - L)\, c_t + U^{t+1}(1, k, u-1, q) \\ g_t + e_t + U^{t+1}(1, k, u, q) \\ U^{t+1}(0, k, u, q) \end{cases}$$

$$U^t(1, k, u, q) = \min \begin{cases} e_t + C\, c_t + U^{t+1}(1, k, u, q-1) \\ e_t + U^{t+1}(1, k, u, q) \\ U^{t+1}(0, k, u, q) \end{cases}$$

where the three terms in the minimisation correspond respectively to producing, setting up the machine without producing, and setting the machine off. For $k = t$, $U^t(y, t, 0, 0)$ can be computed as follows:

$$\begin{aligned} U^t(0,t,0,0) = \quad & \min_{(k>t,u,q)} \ \{g_t + e_t + (d_{t,k-1} - u\,(C-L) - q\,C)\,c_t + U^{t+1}(1,k,u,q) : \\ & \qquad (d_{t,k-1} - u\,(C-L) - q\,C) \leq C - L\} \\ U^t(1,t,0,0) = \quad & \min_{(k>t,u,q)} \ \{e_t + (d_{t,k-1} - u\,(C-L) - q\,C)\,c_t + U^{t+1}(1,k,u,q) : \\ & \qquad (d_{t,k-1} - u\,(C-L) - q\,C) \leq C\} \end{aligned}$$

Then, the optimal solution is

$$\min\{U^1(0,1,0,0)\,,\ \min_{(k>1,u,q)} \{(d_{1,k-1} - u\,(C-L) - q\,C)\,h_0 + U^1(0,k,u,q)\}\}.$$

These computations require $O(T^4)$ time.

5 Computational Results

In this section we present our computational results for the multi-item lot-sizing problem with continuous set-up, stationary start-up times, and Wagner-Whitin costs. For the implementation of the proposed algorithm we used our C-code IPCG – Integer Programming Column Generation (Vanderbeck, 1994). IPCG is a set of subroutines that solves a master integer program using a LP-based branch-and-bound algorithm, where the LP-bounds are obtained using a column generation procedure. The code implements various kinds of branching scheme and termination criteria specific to integer programming column generation algorithms (Vanderbeck and Wolsey 1996, Vanderbeck 1995) and it also allows for the dynamic generation of cutting planes for the master. New applications can be quickly integrated by providing a solver for the column generation subproblem.

In our implementation of the above dynamic programs we construct the set of feasible states at stage k by considering all feasible actions from each of the feasible states at stage $(k-1)$. We use a Red-Black-Tree data structure to maintain the list of feasible states at the current stage; in this way we can test whether the state that we construct already exists in $O(\log l)$ time, where l is the current size of the list, and then update its cost. Due to the significant complexity of the dynamic programming solution of the subproblem, we have also implemented a greedy heuristic procedure that iteratively satisfies demands by assigning production to the period with lowest incremental cost per unit. We solve the subproblem exactly by dynamic programming only when the heuristic fails to return a column with negative reduced cost (the total time used by the subproblem heuristic is typically less than 1% of the total CPU time). The master linear programs are solved using CPLEX 3.0 callable library (CPLEX, 1994). After each solution of the master LP we solve a subproblem for each item. The alternative strategy, that consists in reoptimizing the master every time a column is added, has proved less efficient. The computations have been carried out on an HP9000/712/80 workstation with 64Mb of main memory.

We have generated test problems randomly as follows. First we select a number of items n in $\{3,5\}$, a number of periods T in $\{12,24,36,48\}$, an average capacity C_{aver} in $\{24,30,40\}$. Then, for each item i, we generate the associated capacity C^i from the discrete uniform distribution $U[0.8\,C_{aver},\ 1.2\,C_{aver}]$ and the associated start-up time L^i from the discrete uniform distribution $U[0.2\,C^i,\ 0.6\,C^i]$. Holding, set-up and start-up costs are generated using discrete uniform distributions: $h^i_t \in U[1,3]$, $h^i_0 \in U[300,400]$, $f^i_t \in U[10,12]$, and $g^i_t \in U[20,22]$. Production costs are zero. Hence, costs satisfy the Wagner-Whitin assumption. The demands are generated from $U[0, d_{\max}]$ where $d_{\max}$ is set to $40/n$. Thus, the ratio of the average total machine workload over the average total production capacity

$$\frac{n\,T\,\frac{d_{\max}}{2}}{T\,C_{aver}} = \frac{20}{C_{aver}}$$

is $\frac{5}{6}$, $\frac{2}{3}$, or $\frac{1}{2}$ depending on whether C_{aver} is 24, 30, or 40. For each selection of n, and C_{aver} we generate 5 problem instances (i.e. for each number of periods T, we generate a total of 30 instances), except for problems with 48 periods where we generated 2 instances of each kind.

In Table 4 we report our results for test problems with 36 time periods when the master cuts (31) are not used. The columns contain respectively the instance name (*name*) that includes the number of items; the number of periods; the average capacity and an instance number; the master LP value (*LP*); the optimal solution (*IP*); the depth of the branch-and-bound tree (*depth*); the number of nodes that have been processed in the branch-and-bound tree (*nod*); the number of times the master LP has been solved (*mast*); the number of times the column generation subproblem has been solved (*SP*); the number of columns generated using the heuristic algorithm (*HC*); the total CPU time in seconds (*time*), and the percentage of this time spent in solving subproblems (*tsp*). Table 4 shows how the computational effort varies with the ratio of machine workload over machine capacity and with the number of items.

In Table 5 we give the average results for test problems with 12, 24, 36, and 48 time periods using a branch-and-price algorithm (without master cuts), a branch-and-price-and-cut algorithm (i.e using master cuts of the form (31) at each node of the branch-and-bound tree), and a truncated branch-and-price algorithm (i.e. fathoming nodes for which the gap between lower and upper bounds is less than 2%). In the latter case, we use a rounding heuristic to generate intermediate incumbent solution as described in Vander-

name	LP	IP	depth	nod	mast	SP	HC	time	tsp
n3t36c24p0	6487	6487	0	1	216	648	349	364.4	97.5
n3t36c24p1	12708	12716	1	3	304	909	356	706.9	97.6
n3t36c24p2	16826	16826	0	1	237	711	379	426.0	97.4
n3t36c24p3	18367	18495	2	5	318	951	457	714.7	97.6
n3t36c24p4	20211	21192	7	21	819	2454	480	2079.8	97.7
n3t36c30p0	2473	2473	0	1	221	663	371	228.6	95.9
n3t36c30p1	9330	9330	0	1	245	735	347	349.5	96.6
n3t36c30p2	9731	10207	3	9	516	1542	265	1035.9	97.3
n3t36c30p3	12712	12879	2	5	372	1116	364	709.8	97.1
n3t36c30p4	13540	13540	0	1	221	663	280	361.9	97.3
n3t36c40p0	2403	2408	2	5	360	1077	275	415.9	95.5
n3t36c40p1	7988	7988	1	2	253	756	284	269.3	95.6
n3t36c40p2	8314	8314	0	1	239	717	344	209.8	95.0
n3t36c40p3	8360	8366	2	5	432	1293	298	618.3	95.8
n3t36c40p4	9210	9210	0	1	233	699	371	207.6	95.0
n5t36c24p0	36698	38056	14	123	745	3720	285	1859.4	97.5
n5t36c24p1	46579	47445	8	35	365	1820	260	1061.0	98.2
n5t36c24p2	43445	44979	18	191	1030	5135	337	3161.8	97.8
n5t36c24p3	58633	59298	7	33	262	1310	235	847.0	98.4
n5t36c24p4	36977	37898	10	51	398	1990	319	1072.5	97.7
n5t36c30p0	20180	20758	4	11	258	1285	312	514.9	97.3
n5t36c30p1	24935	25773	9	31	346	1720	338	713.1	97.0
n5t36c30p2	19679	21232	19	143	767	3835	327	1878.9	96.8
n5t36c30p3	28110	29512	14	75	564	2810	290	1300.2	97.3
n5t36c30p4	19463	20113	7	17	273	1365	298	516.4	97.4
n5t36c40p0	15263	15263	0	1	133	665	321	127.2	95.8
n5t36c40p1	15729	15729	0	1	169	845	392	192.3	95.9
n5t36c40p2	13083	13083	0	1	148	740	436	117.9	94.7
n5t36c40p3	14254	14257	1	3	235	1170	504	273.3	95.5
n5t36c40p4	10721	10765	2	5	214	1070	409	231.2	95.6
average	18746.9	19153.0	4.43	26.1	363.1	1480.4	342.7	752.2	96.7

Table 4: Solution of instances with 36 periods.

beck (1996); LP is then the root lower bound which might be lower than the master LP value if the column generation procedure is terminated early at the root node, and IP represents the best integer solution found. The first column of the Table contains the number of periods T and the fraction of instances that were solved for problems with 48 periods; column MC shows the total number of master cuts added during the course of the algorithm.

	branch-and-price (without master cuts)									
T	LP	IP	depth	nod	mast	SP	HC	MC	time	tsp
12	12449.1	12526.4	0.68	2.87	34.1	132.2	65.6	0	1.5	72.7
24	14994.4	15182.0	2.17	8.22	127.3	494.7	186.6	0	50.4	93.7
36	18746.9	19153.0	4.43	26.1	363.1	1480.4	342.7	0	752.2	96.7
48 $(\frac{8}{12})$	13286.1	13452.1	5.00	36.5	736.2	2926.1	559.7	0	3852.5	97.0
	branch-and-price-and-cut (with master cuts)									
T	LP	IP	depth	nod	mast	SP	HC	MC	time	tsp
12	12471.8	12526.4	0.64	2.52	35.2	136.5	65.9	0.44	1.7	72.8
24	15061.1	15182.0	1.70	6.78	128.5	500.9	184.4	4.39	52.8	93.4
36	18787.7	19153.0	3.88	20.5	386.6	1576.6	337.9	26.6	835.6	96.1
48 $(\frac{8}{12})$	13303.6	13452.1	4.13	29.2	762.3	3021.8	553.6	34.0	4056.9	96.5
	truncated branch-and-price (gap $< 2\%$) with rounding heuristic									
T	LP	IP	depth	nod	mast	SP	HC	MC	time	tsp
24	14956.0	15240.4	0.76	3.06	104.8	393.1	184.6	0	33.8	91.5
36	18738.3	19235.9	2.14	9.81	312.7	1224.1	350.2	0	550.3	96.2
48 $(\frac{8}{8})$	13283.7	13473.5	1.13	5.00	492.5	1910.5	557.1	0	2049.5	96.9
48 $(\frac{12}{12})$	17957.7	18427.7	2.67	14.8	735.6	2907.3	604.6	0	3961.1	97.3

Table 5: Average solutions of continuous set-up lot-sizing problems with stationary capacities and stationary start-up times and Wagner-Whitin costs.

Table 5 emphasizes rapidly increasing computational effort with the number of periods. For problems with 48 periods, 4 out of the 12 instances could not be solved to optimality within the allocated 4 hours of CPU time; the result averages are over the solved instances. The remaining sample of 8 instances still includes at least one instance for each combination of n and C_{aver}. The second set of results shows that although the master cuts improve the master LP bound and hence yield smaller a branch-and-bound tree, overall computational effort does not decrease when these cuts are used. The third set of results shows that, for instances with 24 and 36 periods, a solution which is guaranteed to be within 2% of the optimal solution can be obtained in approximately $\frac{3}{4}$ of the time required to obtain an optimal solution. For problems with 48 periods, we first consider the 8 instances that were solved by the exact algorithm for the sake of comparison; we then give average results over the full set of 12 test problems. The rounding heuristic has not been used for the exact algorithm (first two sets of results) because integer solutions naturally arise in the course of the column generation procedure and typically the optimal integer solution is found before the optimal lower bound is obtained. Moreover, the rounding heuristic results in the generation of many more columns.

Finally, we have solved the LP relaxation of the compact formulation $LCST$ (1-8) for comparison purposes. We have added constraints

$$s^i_{t-1} + d^i_t\, y^i_t \geq d^i_t \quad \forall i\, t \tag{37}$$

a priori to $LCST$. Table 6 compares the average (over 30 instances) LP bounds that result from formulations $LCST$ ((1-8) + constraints (37)), M (27-30), and M augmented with cuts (31), for problems with 24 and 36 periods. We also attempted to solve an instance with $n = 3$, $T = 24$, and $C_{aver} = 24$, using the branch-and-bound procedure CPLEXMIP applied to the augmented compact formulation $LCST$, using the same branching priority. With the IP column generation algorithm the problem is solved in 113 seconds and 7 nodes. With CPLEXMIP, the node limit of 20 000 nodes is reached after 995 seconds, leaving an optimality gap of 69%. These figures clearly show the superiority of the column generation approach over a standard LP-based branch-and-bound approach.

nb of periods (T)	24	36
LP relaxation of augmented LCST	6950.05	7175.63
LP gap for augmented LCST	54.22%	62.53%
LP relaxation of M	14994.40	18746.94
LP gap for M	1.23%	2.12%
LP relaxation of M + cuts	15061.14	18787.74
LP gap for M + cuts	0.79%	1.90%
IP	15182.06	19153.07

Table 6: Comparison of the average LP Bounds using the compact and the Master formulation (without and with additional cuts).

6 Final Remarks

In this paper, we have considered a single-machine multi-item lot-sizing model with continuous setups and start-up times. We positioned the problem in the literature and discussed implicit assumptions of alternative ways of modelling start-up times and the complexities of resulting models. The essence of this study has been to show that, even though this problem is complex, instances of realistic size can be solved to optimality. Computational results demonstrate the strength of an integer programming column generation approach (although our implementation is only a prototype) as an alternative to a branch-and-cut approach (such as the one developed by Constantino, 1995 and 1996).

There is scope for improving our code. As more than 90% of the CPU time is used by the dynamic programming procedures, a more efficient implementation of the dynamic programs for the subproblem can lead to significant reduction in overall computational times. We have considered the simplification to the dynamic programming solution of the subproblem that results from Wagner-Whitin costs. Further simplification of the dynamic programming solution to the subproblem could arise from making assumptions on set-up and start-up costs, such as non-increasing costs (Bitran and Yanasse, 1982). Another branching scheme could be used consisting in using SOS branching, i.e. as $\sum_{i \in I} y_t^i \leq 1$ for all t and $I = \{1, \dots, n\}$, one can branch by ensuring that either $\sum_{i \in S} y_t^i \leq 0$ or $\sum_{i \in I \setminus S} y_t^i \leq 0$ for some $S \subseteq I$. This branching scheme should lead to a more balanced tree. Alternatively, one could use a non-binary scheme where branches of the type $y_t^{\bar{\imath}} = 1$ are defined for the various items $\bar{\imath} \in S$ that compete for period t and the branch $\sum_{i \in S} y_t^i \leq 0$ completes the partition. However, for our test problems which involve only a few items, no improvement was gained when experimenting with these alternative branching schemes.

We finish by listing topics for further research. We have seen that defining start-up times as an integer number of periods plus a fraction of a period, i.e. $L = k\,C + \sigma$, with $k \in I\!N$ and $0 \leq \sigma < C$, leads to a more flexible model that is worth studying. On could also relax the single mode assumption by replacing it with the softer assumption that at most one start-up can take place in any time period (Wolsey, 1997). Our computational results show that the intersection of the subproblem convex hulls and the master constraints (including Constantino's cuts (31)) only provides an approximation of the convex hull of the the multi-item lot-sizing solutions. This suggests that further study of the polyhedra of multi-item problem could lead to finding new cuts that involve multiple items. The final question is whether the decomposition approach is a practical approach for the generalization of model LCST to the case of multiple machines.

Acknowledgement
We thank Miguel Constantino for providing appropriate references and Robin Thompson for reviewing our draft. The author is supported in part by a Management Research Fellowship granted by the British Economic and Social Research Council (ESRC).

References

Aggarwal A. and J.K. Park (1993). Improved Algorithms for Economic Lot Size problems. *Operations Research*, Vol. **41**, pp 549-571.

Barany, Van Roy, and Wolsey (1984). Strong formulations for multi-item capacitated lot-sizing. *Management Science*, Vol **30**, pp. 1255-1261.

Bitran G. R. and H.H.Yanasse (1982). Computational Complexity of the Capacitated Lot Size Problem. *Management Science*, Vol **28**, No. 10, pp. 1174-1186.

Cattryse D., M. Salomon, R. Kuik, and L.N. Van Wassenhove (1993). A Dual Ascent and Column Generation Heuristic for the Discrete Lot-sizing and Scheduling Problem with Set-up Times. *Management Science*, Vol. **39**, No. 4, pp 477-486.

Constantino M. F. (1995). A Polyhedral Approach to Production Planning Models: Start-up Costs and Times, Upper and Lower Bounds on Production. Ph.D. Thesis, Département de Mathématique, Université Catholique de Louvain, Louvain-la-Neuve, Belgium.

Constantino M. F. (1996). A Polyhedral Approach to a Production Planning Problem. Research Paper, Departamento de Estatistica e Investigacao Operacional, Universidade de Lisboa.

CPLEX (1994), *Using the CPLEX Linear Optimizer*, version 3.0, CPLEX Optimization, Inc., Suite 279, 930 Tahoe Blvd., Bldg. 802, Incline Village, NV 89451-9436. (702) 831-7744.

du Merle O., J.-L. Goffin, C. Trouiller, and J.-P. Vial (1997). A Lagrangian Relaxation of the Capacitated Multi-Item Lot Sizing Problem solved with an Interior Point Cutting Plane Algorithm. Les Cahiers du GERAD, April 1997.

Federgruen A. and M. Tzur (1991). A Simple Forward Algorithm to Solve General Dynamic Lot Sizing Models with n Periods in $O(n \log n)$ or $O(n)$ Time. *Management Science*, Vol **37**, No. 8, pp. 909-925.

Fleischmann B. (1990). The discrete lot-sizing and scheduling problem. *European Journal of Operational Research*, **44**, pp. 337-348.

Florian M., J.K. Lenstra, and A.H.G. Rinnooy Kan (1980). Deterministic Production Planning: Algorithms and Complexity. *Management Science*, Vol. **26**, No. 7, pp. 669-679.

Florian M. and M. Klein (1971). Deterministic Production Planning with Concave Costs and Capacity Constraints. *Management Science*, Vol. **18**, No. 1, pp. 12-20.

Leung, Magnanti and Vachani (1989). Facet and algorithms for capacitated lot sizing. *Mathematical programming*, Vol. **45**, pp 331-359.

Magnanti T.L. and Vachani R. (1990). A Strong Cutting Plane Algorithm for Production Scheduling with Changeover Costs. *Operations Research*, Vol. **38**, No. 3, pp 456-473.

Nemhauser G.L. and L.A. Wolsey (1988). *Integer and Combinatorial Optimization.* John Wiley & Sons, Inc.

Pochet Y., and L.A. Wolsey (1993). Lot-sizing with Constant Batches: Formulation and Valid Inequalities. *Mathematics of Operations Research*, Vol. **18**, No. 4, pp 767-785.

Pochet Y., and L.A. Wolsey (1995). Algorithms and Reformulations for Lot Sizing Problems. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. **20**.

Salomon M., L.G. Kroon, R. Kuik, and L.N. Van Wassenhove (1991). Some Extensions of the Discrete Lot-sizing and Scheduling Problem. *Management Science*, Vol. **37**, No. 7, pp 801-812.

Trigeiro W. W., L.J. Thomas, and J. O. McClain (1989). Capacitated Lot Sizing with Set-up Times. *Management Science*, Vol. **35**, No. 3, p. 353-366.

van Hoesel S. and A. Kolen (1994). A linear description of the discrete lot-sizing and scheduling problem. *European Journal of Operational Research*, **75**, pp. 342-353.

van Hoesel C. P. M. and A. P. M. Wagelmans (1996). An $O(T^3)$ Algorithm for the Economic Lot-sizing Problem with Constant Capacities. *Management Science*, Vol. **42**, No. 1, p. 142-150.

Vanderbeck F. (1994). Decomposition and Column Generation for Integer Programs, Ph.D. Thesis, Faculté des Sciences Appliqées, Université Catholique de Louvain, Louvain-la-Neuve, Belgium.

Vanderbeck F. and L. A. Wolsey (1996). An Exact Algorithm for IP Column Generation, *Operations Research Letters* Vol. **19**, No 4, pp 151-159.

Vanderbeck F. (1995). On Integer Programming Decomposition and ways to enforce integrality in the Master, *Research Papers in Management Studies*, University of Cambridge, 1994-1995, no 29 (revised in May 1996).

Vanderbeck F. (1996). Computational Study of a Column Generation algorithm for Bin Packing and Cutting Stock problems, *Research Papers in Management Studies*, University of Cambridge, 1996 no 14.

van Eijl Cleola (1996). A Polyhedral Approach to the Discrete Lot-sizing and Scheduling Problem, Ph.D. Thesis, Technische Universiteit Eindhoven, Netherlands.

Wagelmans A., S. van Hoesel and A. Kolen (1996). Economic Lot Sizing: An $O(n \log n)$ Algorithm that Runs in Linear Time in the Wagner-Whitin Case. *Operations Research*, Vol. **40**, S145-S156.

Wolsey L. A. (1997). MIP modelling of changeovers in production planning and scheduling problems. To appear in the European Journal of Operations Research.