DEPARTEMENT TOEGEPASTE ECONOMISCHE WETENSCHAPPEN

ONDERZOEKSRAPPORT NR 9610

Clips: a capacity and lead time integrated procedure for scheduling

by

Marc R. LAMBRECHT

Philip L. IVENS

Nico J. VANDAELE

Katholieke Universiteit Leuven

Naamsestraat 69, B-3000 Leuven

ONDERZOEKSRAPPORT NR 9610

Clips: a capacity and lead time integrated procedure for scheduling

by

Marc R. LAMBRECHT
Philip L. IVENS
Nico J. VANDAELE

D/1996/2376/10

CLIPS: A CAPACITY AND LEAD TIME INTEGRATED PROCEDURE FOR SCHEDULING *

MARC R. LAMBRECHT, PHILIP L. IVENS, NICO J. VANDAELE
Department of Applied Economics
Katholieke Universiteit Leuven
Naamsestraat 69, B-3000 Leuven (Belgium)
Phone : 32-16-326971, Fax : 32-16-326732
E-mail : Marc.Lambrecht@econ.kuleuven.ac.be
E-mail : Philip.Ivens@econ.kuleuven.ac.be
E-mail : Nico.Vandaele@econ.kuleuven.ac.be

January 1996

Abstract

We propose a general procedure to address real life job shop scheduling problems. The shop typically produces a variety of products, each with its own arrival stream, its own route through the shop and a given customer due date. The procedure first determines the manufacturing lot sizes for each product. The objective is to minimize the expected lead time and therefore we model the production environment as a queueing network. Given these lead times, release dates are set dynamically. This in turn creates a time window for every manufacturing order in which the various operations have to be sequenced. The sequencing logic is based on an Extended Shifting Bottleneck Procedure. These three major decisions are next incorporated into a four phase hierarchical operational implementation scheme. A small numerical example is used to illustrate the methodology. The final objective however is to develop a procedure that is useful for large, real life shops. We therefore report on a real life application.

*This research was supported by the National Science Foundation (NFWO/FKFO Belgium), project 2.0053.93

1 Introduction

The production environment modelled in this paper, can best be described as a multi-operation job shop scheduling problem under an assemble(make)-to-order customer policy. Customers arrive dynamically with a request for work. Each customer order is characterized by a certain volume and mix and an agreed due date. There is no inventory of finished products, so the orders placed in the shop enter a backlog, or queue of orders. Each order requires several operations on different machines, the routes which are characterized by a bill of processes are not necessarily the same for each order. We moreover explicitly include the stochastic nature of the production system. The stochastic nature is on the level of the customer orders itself (the shop typically produces a large variety of products each with its own stochastic arrival stream) and on the level of the shop floor, where processing and setup times are not deterministic due to all sorts of variability and disruptions. In this paper we propose a general procedure to address this problem. It is clear that this problem has attracted a great deal of attention from many researchers. The situation we have in mind, is very similar to the one described by Wein and Chevalier [20], although we propose a different methodology.

The hierarchical approach we propose consists of three important decisions. The first decision can be best described as a lot sizing decision. Individual customer orders for the same product are grouped into manufacturing orders. The lot sizing policy is a typical fixed order quantity rule, grouping customer orders in preferably small lots which are frequently produced. The lot sizes are the outcome of a queueing model. The whole production system is modelled as a queueing network, in which all operations (and its parameters) and arrival streams are stochastically represented. The outcome of this exercise are lot sizes per product which minimize the expected lead time. There are many multi-product, multi-machine queueing networks described in the literature (e.g. Leung and Suri [18]). We developed our own network approach where all parameters are a function of the lot size (see section 2). The queueing approach has many advantages, it explicitly includes the convex relationship between lot sizes and lead times (see e.g. Karmarkar [14] and Lambrecht and Vandaele [17]), it takes care of congestion phenomena (the impact of the utilization of the most heavily loaded machines), it quantifies the queueing delays and it takes into account the stochastic nature of the problem. The result of the queueing network are target lot sizes (of manufacturing orders) which give an indication of how customer orders have to be grouped. We just described our lot sizing rule as a fixed order quantity rule. This is not accurate. We group customer orders in such a way that we approach the target lot sizes as close as possible. Given the time varying nature of the customer demands, the manufacturing orders may actually differ from order to order, but on the average we aim for lot sizes minimizing the expected lead time (and work-in-process). Given this grouping of real customer orders, we characterize our model rather as a make-to-order policy than a make-to-stock approach.

The second major decision is the determination of the release date of the manufacturing orders. The release date is set equal to the due date minus the lead time estimate of the manufacturing order. The estimate of the lead time is equal to the expected lead time plus a safety lead time. The safety lead time depends on the customer service. The lead time estimate is such that we expect to satisfy customer orders timely P % of the time. The question is what amount of safety time do we have to add to the expected lead time (for every manufacturing order) so that a customer service of P % is obtained. This of course requires knowledge of the probability distribution of the lead time. Therefore, an estimate of the variance of the lead time is mandatory. This variance is also an outcome of our queueing model. It is interesting to note that we start from a given due date and that, consequently, we do not assign due dates to exogenously arriving jobs. This approach is different from the one adopted by Wein and Chevalier [20].

The third major decision concerns the sequencing policy. In the previous step a time window is created for every manufacturing order. This time window is the estimate of the lead time (expected lead time plus safety time). Within these time windows (one for every manufacturing order) we now have to sequence all operations in detail. We opted for the shifting bottleneck procedure (Adams, Balas and Zawack [1]) for various reasons, one being its excellent performance as described by Ivens and Lambrecht [13]. The shifting bottleneck procedure has to be adopted so that it can be used to sequence the operations for our general job shop environment including assembly operations, release dates, due dates, overlapping operations, multiple resources (machines and labour force), setup times, calendars and many other real life features. The ESBP (Extended Shifting Bottleneck procedure) is described in section 3.

This methodology, based on three major decisions (the lead time estimation, the releasing decision and the sequencing policy), is next transformed into a hierarchical, four phase, operational implementation scheme. This implementation scheme is summarized in figure 1.

The four phases of the implementation can be summarized as follows. Phase one is the lead time estimation and lot sizing step. In this phase the manufacturing system is transformed into a queueing network (based on customer orders, the various resource types, calendars, routings, etc.). The outcome are lot sizes and lead time estimations. The second step is a tuning phase. Management may consider the lead times as unacceptable. To remedy the situation, management may decide to adjust the capacity structure (e.g. overtime), to off-load heavily loaded resources, to consider alternative routings or to reassign new due dates. The adjustments may result in a new run of the queueing model. The actions to be taken here depend upon the practical situation on hand. The next phase is the scheduling phase, including (a) the grouping of customer orders into manufacturing orders (b) determining the release date for each manufacturing order (which is set equal to the due date minus the lead time estimate) and (c) the

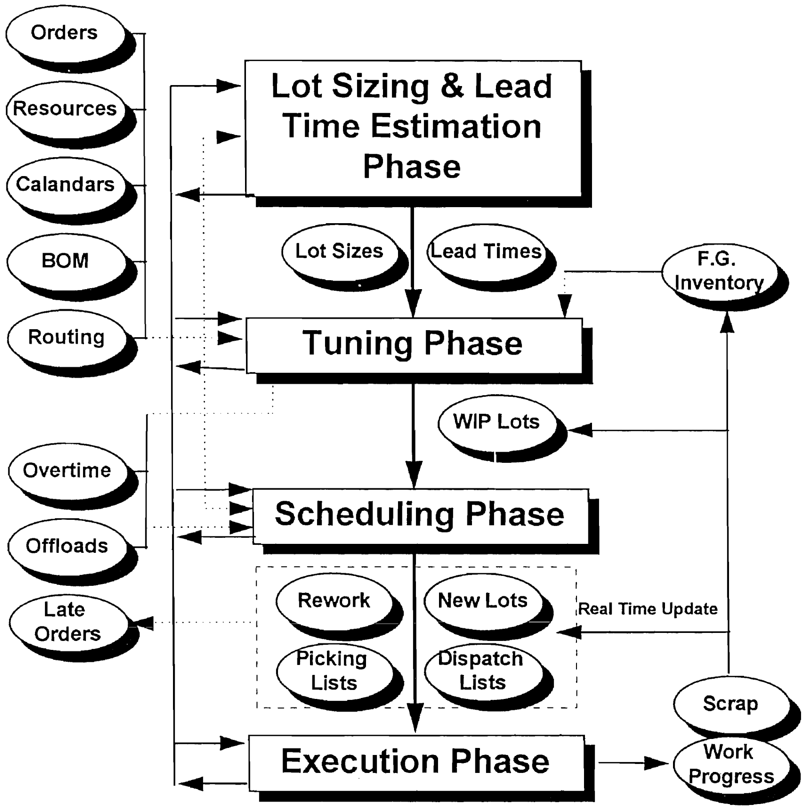

Figure 1: The four phase hierarchical approach of CLIPS

detailed sequencing of all operations. In the final phase, the detailed plans are transferred to the shop floor on a real-time basis. Through electronic data captation, information concerning the execution of the detailed plan, is fed back so that rescheduling can be done. The nature and frequency of rescheduling heavily depends on the dynamics of the real life situation and the level of responsiveness required.

The system described above was named CLIPS (Capacity and Lead Time Integrated Procedure for Scheduling) and has been fully implemented in a metal working company. The applications will be described in section 4.

The major advantages of our approach are the following: a simultaneous treatment of the capacity and material flow in every step of the procedure. Lead times are estimated through a queueing model taking into account congestion phenomena and the queueing impact of lot sizing. The estimation of a time window for sequencing (release date, due date) includes a safety margin so that customer service targets can be specified. The tuning phase allows a management intervention to manage the capacity/inventory (lead time) trade-off. The scheduling phase copes with the complexity of detailed scheduling while the dynamics of the floor are captured in the execution phase.

The remainder of the paper is organized as follows. The lead time estimation and lot sizing phase is discussed in section 2. The scheduling phase is explained in section 3. We use a small example to illustrate the algorithms. This small example is only used for illustrative purposes and does not reflect the full potential of the methodology to tackle real life problems. To get a better idea of this potential we discuss a real life application in section 4. We draw conclusions in section 5.

2 Lead time estimation and lot sizing phase

2.1 Introduction

In CLIPS, the production environment is modelled as a network of queues. In this analytic approach, equations (for the expected lead time and the variance of the lead time) are derived that capture the dynamics of the system in an aggregate way. The arrival process for each product is characterized by the average customer demand quantities and the average and variance of the order interarrival times. The exogenous arrival rate can be derived from historical data or from demand forecasts depending on the availability of data. The other parameters are: the service times (average and variance of both setup and unit processing times) and shop parameters such as routings and calendars. The outcome of the model are explicit functions for the expected lead time and variance of the lead time as a function of the lot size. We assume a constant lot size per product over the entire routing. Next an optimization routine is used to find the lot sizes that minimize the expected lead time. We call these lot sizes 'target lot sizes'. A lognormal distribution is postulated to characterize the lead time distribution. This in turn allows the user to specify a lead time, satisfying a predetermined customer service.

Throughout the paper a small example will be used to numerically illustrate the various steps of our procedure. The shop, a small metal shop, is shown in figure 2. The metal shop fabricates two products, P and S and has three machine (centers) types: a cutter (C), a grinder (G) and a lathe (L). Product P has three stages on its route (on machine C, G and L) and product S has two stages (on machine L and G). The shop runs three shifts per day, seven days a week. There is one machine available of each type. The customer demands (we only show the demands for next month) are summarized in table 1. As can be seen, we have 5 customer orders for product P and 15 orders for product S. Each order is characterized by an order quantity, a due date and an interarrival time.

Table 1 has to be interpreted as follows: one unit of product P has to be delivered at day 22, 5 units at day 28, 3 units at day 37, etc. The order interarrival times can be derived easily (we assume that we are currently on day 14). The

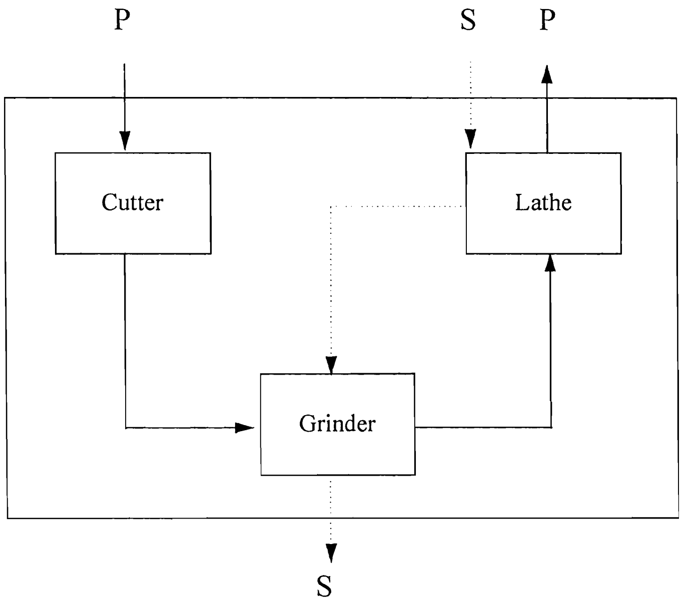

Figure 2: The layout of the small metal shop

processing and setup times are summarized in table 2 (all times are expressed in hours).

In table 2 it can be seen that both the cutter and the lathe have deterministic setup and processing times. The grinder faces exponentially distributed setup and unit processing times, because this operation has to be constantly monitored and adjusted by a worker.

2.2 Model derivation

We will now discuss the formal treatment of the lead time estimation and lot sizing phase. Assume k to be the product index ($k = 1 \ldots K$), m the machine index ($m = 1 \ldots M$) and o the operation index for product k ($o = 1 \ldots O_k$), where O_k is the number of operations for product k. For each product k we have C_k orders (customer orders or forecasts). Each order c for product k ($1 \leq c \leq C_k$) is characterized by an order quantity OQ_{kc} and an order due date DD_{kc}. From the sequence of order due dates, the order interarrival times Y_{kc} can easily be derived

$$Y_{kc} = DD_{kc} - DD_{k\,c-1} \tag{1}$$

which is, for product k, the elapsed time between order $c-1$ and order c. Other characteristics of the order arrivals can be derived

$$\overline{Y}_k = \frac{1}{C_k} \sum_{c=1}^{C_k} Y_{kc} \tag{2}$$

$$
\begin{array}{llllll}
\overline{Y}_P & = & 144 & \overline{Y}_S & = & 48 \\
s^2_{Y_P} & = & 3744 & s^2_{Y_S} & = & 494 \\
c^2_{Y_P} & = & 13/72 & c^2_{Y_S} & = & 3/14 \\
\lambda_P & = & 1/144 & \lambda_S & = & 1/48 \\
\overline{OQ}_P & = & 3 & \overline{OQ}_S & = & 2
\end{array}
$$

As far as the production characteristics are concerned, the following is defined for product k and operation o, expressed in hours:

T_{ko}	setup time random variable
X_{ko}	unit processing time random variable
$\overline{T}_{ko}$	expected setup time
$\overline{X}_{ko}$	expected unit processing time
μ_{ko}	unit processing rate ($= 1/\overline{X}_{ko}$)
$s^2_{T_{ko}}$	variance of the setup time
$s^2_{X_{ko}}$	variance of the unit processing time
$c^2_{T_{ko}}$	scv of the setup time
$c^2_{X_{ko}}$	scv of the unit processing time

In addition, define

$$
\begin{array}{llll}
\delta_{kom} & = & 1 & \text{if operation } o \text{ for product } k \text{ is on machine } m \\
 & = & 0 & \text{otherwise}
\end{array}
$$

The routing of the metal shop gives us

$$
\begin{array}{lll}
\delta_{P1C} = 1 & \delta_{P2G} = 1 & \delta_{P3L} = 1 \\
\delta_{S1L} = 1 & \delta_{S2G} = 1 &
\end{array}
$$

and all other δ_{kom}'s equal 0.

A this point all the input parameters are given. We use a queueing network approach to model the job shop. The job shop is viewed as a network of machines which are linked by the various flows (routings). Each machine is modeled as a multi product lot sizing model with queueing delays. The multiple arrival processes of the k products are superposed into one aggregate arrival process. All characteristics of the aggregate arrival process and the aggregate production process are functions of the lot sizes Q_k. Note that we express in our derivations Q_k as a multiplier of the average order quantity $\overline{OQ}_k$. For each machine m we have to obtain:

λ_{ba_m}	the aggregate batch arrival rate
$c^2_{ba_m}$	the scv of the aggregate batch interarrival time
$c'^2_{ba_m}$	the scv of the external aggregate batch interarrival time
μ_{bs_m}	the aggregate batch processing rate
$c^2_{bs_m}$	the scv of the aggregate batch processing time
ρ'_m	the adapted traffic intensity

the proportion of batches leaving machine m and going outside, $\mathbf{F}$, the transition matrix of f_{mn} $(m, n = 0 \ldots M)$.

Solving the following set of linear equations yields the M unknowns $c^2_{ba_m}$, $m = 0, \ldots, M$:

$$-\sum_{n=1}^{M} \lambda_{ba_n} f^2_{nm} (1 - \rho'_n) c^2_{ba_n} + \lambda_{ba_m} c^2_{ba_m} =$$

$$\sum_{n=1}^{M} \lambda_{ba_n} f_{nm} (f_{nm} {\rho'_n}^2 c^2_{bs_n} + 1 - f_{nm}) + \lambda'_{ba_m} {c'}^2_{ba_m} \tag{15}$$

Equations (15) are a slightly adapted version (in terms of general exogenous arrivals instead of Poisson arrivals) of the results obtained by Shantikumar and Buzacott [19]. The entrances of the transition matrix $\mathbf{F}$ are obtained as follows:

$$f_{0n} = \frac{\lambda'_{ba_n}}{\sum_{m=1}^{M} \lambda'_{ba_m}} \tag{16}$$

$$f_{mn} = \frac{1}{\lambda_{ba_m}} \sum_{k=1}^{K} \sum_{l=1}^{Q_k - 1} \lambda_{b_k} \delta_{kom} \delta_{ko+1n} \tag{17}$$

$$f_{m0} = \frac{1}{\lambda_{ba_m}} \sum_{k=1}^{K} \lambda_{b_k} \delta_{k\, O_k\, m} \tag{18}$$

for $n = 1 \ldots M$ and $m = 1 \ldots M$. Note that in our model, due to the fact that the routings are given, we face deterministic routing. Therefore, the transition matrix $\mathbf{F}$ can be derived in the way described above.

Returning to the small metal shop we have the following transition matrix $\mathbf{F}$:

	0	C	G	L
0	0	$\frac{48Q_S}{48Q_S+144Q_P}$	0	$\frac{144Q_P}{48Q_S+144Q_P}$
C	0	0	1	0
G	$\frac{144Q_P}{48Q_S+144Q_P}$	0	0	$\frac{48Q_S}{48Q_S+144Q_P}$
L	$\frac{48Q_S}{48Q_S+144Q_P}$	0	$\frac{144Q_P}{48Q_S+144Q_P}$	0

To obtain ${c'}^2_{ba_m}$ we use the following approximation if $\sum_{k=1}^{K} \delta_{k1m} \geq 2$

$${c'}^2_{ba_m} \approx \frac{1}{3} + \frac{2}{3} \sum_{k=1}^{K} \pi_{mk1} \frac{c^2_{Y_k}}{Q_k} \tag{19}$$

where

$$\pi_{mk1} = \frac{\lambda_{b_k}\delta_{k1m}}{\sum_{k=1}^{K}\lambda_{b_k}\delta_{k1m}} = \frac{\lambda_{b_k}\delta_{k1m}}{\lambda_{ba_m}} \tag{20}$$

If $\sum_{k=1}^{K}\delta_{k1m} = 1$ then

$$c'^2_{ba_m} = \frac{c^2_{Y_k}}{Q_k} \tag{21}$$

The approximation for $c'^2_{ba_m}$ is the sum of a constant and a weighted average of the scv's of all the external batch arrivals at machine m. It is an interpolation between complete deterministic arrivals (where the aggregate scv is approximated by the scv of a uniform distribution $U[0, 2/\lambda'_{ba_m}]$ and the scv of Poisson arrivals (where all scv's equal one). The lather is the only known exact result in the literature for the superposition of arrival processes. The weights 1/3 and 2/3 in the expression for $c'^2_{ba_m}$ are a particular instance of the general approximation described by Albin [2], [3]. For our illustrative example we obtain

$$\begin{aligned} c'^2_{ba_C} &= \tfrac{13}{72}\tfrac{1}{Q_P} \\ c'^2_{ba_G} &= 0 \\ c'^2_{ba_L} &= \tfrac{3}{14}\tfrac{1}{Q_S} \end{aligned}$$

Then finally the lead time for product k on machine m for operation o

$$E(W_{ko}) = \sum_{m=1}^{M} E(Wq_m)\delta_{kom} + \overline{T}_{ko} + Q_k\overline{OQ}_k\overline{X}_{ko} \tag{22}$$

with

$$\begin{aligned} E(Wq_m) &= \tfrac{\rho'^2_m(c^2_{ba_m}+c^2_{bs_m})}{2\lambda_{ba_m}(1-\rho'_m)} \exp\left\{\tfrac{-2(1-\rho'_m)(1-c^2_{ba_m})^2}{3\rho'_m(c^2_{ba_m}+c^2_{bs_m})}\right\} && \text{if } c^2_{ba_m} \le 1 \\ E(Wq_m) &= \tfrac{\rho'^2_m(c^2_{ba_m}+c^2_{bs_m})}{2\lambda_{ba_m}(1-\rho'_m)} && \text{if } c^2_{ba_m} > 1 \end{aligned}$$

The aggregated objective function for machine m can be stated as follows

$$E(W_{M_m}) = E(Wq_m) + \sum_{k=1}^{K}\pi'_{mk}\left(\sum_{o=1}^{O_k}\frac{\lambda_k\overline{OQ}_k\delta_{kom}}{\lambda'_{mk}}[\overline{T}_{ko} + Q_k\overline{OQ}_k\overline{X}_{ko}]\right)$$

where

$$\pi'_{mk} = \frac{\lambda'_{mk}}{\lambda'_m} \tag{23}$$

with

$$\lambda'_{mk} = \sum_{o=1}^{O_k} \lambda_k \overline{OQ}_k \delta_{kom} \tag{24}$$

and

$$\lambda'_{m} = \sum_{k=1}^{K}\sum_{o=1}^{O_k} \lambda_k \overline{OQ}_k \delta_{kom} \tag{25}$$

This objective function for machine m is the weighted average over the products visiting machine m, which on their turn are weighted averages over the operations on machine m for product k. Note that weight π'_{mk} is independent from the manufacturing lot size. It measures the relative importance of product k for machine m. This is in contrast to the weights for instance used by Baker [5], who uses weights which are function of the manufacturing lot size. Because the manufacturing lot size is a decision variable, these weights can turn to zero for large lot sizes so that the respective term in the objective function can grow to infinity without penalty. This observation led us to remove the decision variable from the weights.

The objective function for the total job shop becomes

$$\begin{aligned} E(W) &= \sum_{m=1}^{M} E(Wq_m) + \sum_{k=1}^{K} \pi'_k \frac{[Q_k\overline{OQ}_k - 1]\overline{Y}_k}{2\overline{OQ}_k} + \\ &\quad \sum_{m=1}^{M}\sum_{k=1}^{K} \pi'_{mk} \left(\sum_{o=1}^{O_k} \frac{\lambda_k \overline{OQ}_k \delta_{kom}}{\lambda'_{mk}} [\overline{T}_{ko} + Q_k \overline{OQ}_k \overline{X}_{ko}] \right) \end{aligned} \tag{26}$$

where

$$\pi'_k = \frac{\lambda_k \overline{OQ}_k}{\sum_{k=1}^{K}\sum_{o=1}^{O_k} \lambda_k \overline{OQ}_k} \tag{27}$$

The latter weight takes care of the relative importance of product k for the total job shop. The second term of equation (26) measures the average waiting time finished batches have to wait until their due date. For the metal shop the objective function for the entire job shop equals

$$\begin{aligned} E(W) &= E(W_{q_C}) + E(W_{q_G}) + E(W_{q_L}) + 20 + 90Q_P \\ &\quad + \frac{1}{3}(20 + 30Q_P) + \frac{2}{3}(20 + 20Q_S) + \frac{1}{3}(24 + 36Q_P) + \\ &\quad \frac{2}{3}(16 + 16Q_S) + 8(3Q_P - 1) + 8(2Q_S - 1) \end{aligned}$$

At this point the formulation of the job shop is complete. Remark that in contrast to traditional queueing networks, our approach includes some new approximations and that every quantity is an explicit function of the manufacturing lot size.

2.3 Optimization and decomposition

The minimization problem involves a non-linear objective function and a set of simultaneous non-linear constraints. A dedicated optimization routine has been developed and is discussed in Lambrecht, Ivens and Vandaele [16]. The optimal lot sizes for the small metal shop are (rounded) $Q_P^* = 4$ and $Q_S^* = 6$ while the aggregate objective function value of the job shop equals 501. The decomposition, after the optimization, can be summarized as follows. The optimal lot sizes Q_k^* (or the vector $\mathbf{Q}^*$) for each product are used to calculate the expected lead time of operation o of product k on machine m

$$E(W_{ko}) = \sum_{m=1}^{M} E(Wq_m(\mathbf{Q}^*))\delta_{kom} + \overline{T}_{ko} + Q_k^*\overline{OQ}_k\overline{X}_{ko} \tag{28}$$

The first term is clearly common for all products using machine m. The total lead time of product k (for the whole routing) is given by

$$\begin{aligned} E(W_k) &= \sum_{o=1}^{O_k} E(W_{ko}) = \frac{Q_k(\overline{OQ}_k - 1)\overline{Y}_k}{2\overline{OQ}_k} + \\ &\sum_{o=1}^{O_k}\sum_{m=1}^{M} E(Wq_m(\mathbf{Q}^*))\delta_{kom} + \sum_{o=1}^{O_k}\overline{T}_{ko} + \sum_{o=1}^{O_k} Q_k^*\overline{OQ}_k\overline{X}_{ko} \end{aligned} \tag{29}$$

product	optimal lot size	operation	adapted traffic intensity (%)	waiting time	setup time	processing time	lead time
P	5	cutter	73	7	20	120	147
		grinder	87	109	20	40	169
		lathe	82	42	24	48	114
		stock					72
		total					502
S	6	lathe	87	42	16	48	106
		grinder	82	109	20	60	189
		stock					60
		total					355

Table 3: Optimal lot size and lead time for the metal shop

This is illustrated in table 3. From this table it can be seen that there is a small queue in front of the cutter. On the other hand, both the grinder and the lathe face long waiting times compared to their processing times. This is mainly due to the high adapted traffic intensities. The waiting time for the grinder is even larger. This is obviously due to the stochastic nature of that facility. The operation 'stock' is the average time that the finished goods from a completed manufacturing batch have to wait until their due date.

The variance of the total lead time of product k equals

$$V(W_k) = \frac{Q_k\overline{OQ}_k - 1}{2\overline{OQ}_k^2} s_{Y_k}^2 + \frac{(Q_k\overline{OQ}_k - 1)(Q_k\overline{OQ}_k + 1}{12\overline{OQ}_k^2}\overline{Y}_k^2 + \sum_{o=1}^{O_k} V(Wq_m)\delta_{kom} + \sum_{o=1}^{O_k} s_{T_{ko}}^2 + \sum_{o=1}^{O_k} Q_k\overline{OQ}_k s_{X_{ko}}^2 \qquad (30)$$

The term $V(Wq_m)$ is derived in appendix. For our example, the standard deviation of the total lead time is 158 for product P and 154 for product S which suggests that the lead times are highly variable.

If the lognormal distribution is assumed, then the parameters are

$$\beta_k = \ln\left(\frac{E(W_k}{\sqrt{\frac{V(W_k)}{E(W_k)^2} + 1}}\right) \qquad (31)$$

$$\gamma_k^2 = \ln\left(\frac{V(W_k)}{E(W_k)^2} + 1\right) \qquad (32)$$

The lead times, including safety time, are obtained in the following way. W_{P_k} is the total lead time guaranteeing a service of P_k%. This means that the manufacturer will satisfy this lead time P_k% of the time for product k. Then

$$W_{P_k} = \exp\{\beta_k + z_{P_k}\gamma_k\} \qquad (33)$$

where z_{P_k} can be obtained from the standard normal table (P_k is the required percentile for product k). In the sequel, we will call W_{P_k} the planned lead time, because it will be used to fix the planned release date. For our example we obtain (for some values of P_k)

P_k	80%	90%	95%	99%
Product P	621	710	794	980
Product S	463	554	644	855

3 Scheduling Phase

In the scheduling phase, basically three types of decisions have to be taken. First, customer orders have to be grouped into manufacturing orders, approaching the previously calculated target lot sizes as close as possible. Next, we have to establish a release date for each manufacturing order. Finally all non-completed operations of both newly released and in-process manufacturing orders have to be sequenced on the different machines. We briefly discuss each of these three steps.

3.1 Grouping of Customer Orders into Manufacturing Orders

The problem addressed here is the grouping of C_k customer orders of product k, characterized by a quantity OQ_{kc} $(1 \leq c \leq C_k)$ and a due date DD_{kc} $(1 \leq c \leq C_k)$, into a number of manufacturing orders $L_{kl}(l = 1, \ldots, S_k)$ of which the number of units ideally approach the previously fixed target lotsize Q_k^*.

For each product k, we first fix the number of setups

$$S_k = \left\lfloor \frac{1}{Q_k^*} \sum_{c=1}^{C_k} OQ_{kc} \right\rfloor \tag{34}$$

where $\lfloor \cdot \rfloor$ is the largest integer smaller than or equal to $\cdot$. In our case S_P equals 3 ($\lfloor 15/4 \rfloor$) and S_s equals 5 ($\lfloor 30/6 \rfloor$). From table 1 we learn that there are 5 customer orders for product P and 15 customer orders for product S. The grouping into manufacturing lots can be done in several ways. It is clear that this problem can be formulated as an integer programming model or, more elegantly, transformed into a dynamic programming formulation. Given the standard nature of this problem we omit the formulation. It is however important to mention that the objective function we used minimizes the number of inventory-days. In table 4 we summarize the results for our illustrative case where QL_{kl} stands for the lot size of the new manufacturing orders.

product	L_{kl}	Grouped Customer orders	QL_{kl}
P	L_{P1}	1-2	6
	L_{P2}	3-4	5
	L_{P3}	5	4
S	L_{S1}	1-2-3	6
	L_{S2}	4-5-6	5
	L_{S3}	7-8	5
	L_{S4}	9-10-11-12	9
	L_{S5}	13-14-15	5

Table 4: The manufacturing lot sizes

3.2 Release of New Manufacturing Orders

For the newly determined manufacturing order quantities, QL_{kl}, we have to compute the corresponding expected lead time and the planned lead time (which includes safety time). We therefore use the equations (29), (30) and (33) which are derived in section 2.3. Next, the planned lead times are deducted from the due dates to obtain the release dates for each manufacturing order. These results are summarized in table 5 (we plan for a 95% service level). The due date for L_{P1} is day 22 (it includes customer orders 1 and 2) so the due date in hours equals 528 from now.

Manufacturing Lot	Due Date (hours)	Expected Lead Time (hours)	Planned Lead Time (hours)	Release Date (hours)
L_{P1}	528	534	581	-53
L_{P2}	888	482	534	354
L_{P3}	1056	430	488	568
L_{S1}	408	295	380	28
L_{S2}	528	277	368	160
L_{S3}	648	277	368	280
L_{S4}	792	349	422	370
L_{S5}	936	277	368	568

Table 5: Release Dates of the Manufacturing Orders

3.3 Detailed Scheduling of the Operations

At this stage of our procedure, all non-completed operations of manufacturing orders are scheduled between the release date (or the current moment if the order is overdue) and the due date of the order. Detailed scheduling requires to specify for each operation of each manufacturing order $L_{kl}(k = 1, \ldots K, l = 1, \ldots S_k)$ when it has to be performed and by what resource, explicitly taking into account the limited availability of the various resources and many other constraints such as precedence among operations, release dates and due dates. A schedule needs to optimize a predetermined objective. Many production managers strive for due date performance, short lead times and low in-process inventory levels.

The well known job shop scheduling problem is the theoretical abstraction of this problem and has been subject of numerous research efforts. Both optimal and heuristic solution procedures are proposed in the literature. Recent integer programming based models can be found in Balas [6] and Applegate and Cook [4]. Among others, Lageweg, Lenstra and Kan [15], Carlier and Pinson [11], Brücker, Jürisch and Sievers [9] and Brücker, Jürisch and Krämer [8] propose implicit enumeration methods for solving the job shop problem. The problem can be stated as follows. N operations have to be scheduled on M resources. Each operation requires a particular resource. A resource can process only one operation at a time and preemption of processing is not allowed. Precedence constraints among operations may exist (e.g. between operations of the same order). A schedule has to be found so that the makespan is minimal. The job shop scheduling problem can be formalized as follows

$$\begin{array}{lll} \min & C & \\ \text{Subject to} & C \geq t_i + p_i & \forall i \in N \\ & t_j \geq t_i + p_i & \forall (i,j) \in A \\ & t_i \geq t_j + p_j \vee t_j \geq t_i + p_i & \forall (i,j) \in E_m, \forall m \in M \\ & t_i \geq 0 & \forall i \in N \end{array}$$

with C, the makespan of the schedule, M, the set of all available resources, N, the set of all operations to be scheduled, p_i, the processing time of operation i, t_i, the starting time of operation i, A, the set of all pairs of operations (i,j) for which i has to precede j, E_m, the set of all pairs of operations that require the same resource m, Unfortunately, the job shop scheduling problem is NP-hard in the strong sense. This implies that there is little hope to find optimal solutions to large real-life scheduling problems within reasonable computer time. For practical applications heuristic schedule generation procedures with priority dispatching rules are often used. Well known dispatching rules are, FCFS (First Come First Served), SPT (Shortest Processing Time), EDD (Earliest Due Date), MWR (Most Work Remaining), CR (Critical Ratio), to mention only a few.

Adams, Balas and Zawack [1] introduced the Shifting Bottleneck Procedure (SBP), a new powerful heuristic for the job shop scheduling problem. Extensions to the SBP, such as Dauzère-Péres and Lasserre [12] and Balas, Lenstra and Vazacopoulos [7] increase its performance. Experiments by Adams, Balas and Zawack [1], Dauzère-Péres and Lasserre [12], Ivens and Lambrecht [13] and Balas, Lenstra and Vazacopoulos [7] indicate that SBP offers exceptionally good results compared to other heuristics such as priority dispatching rules. Because of the SBP's good balance between computational complexity and the quality of the generated schedules, we have chosen this procedure as the engine of our detailed scheduling phase.

However, the scope of the theoretical job shop scheduling problem is far too limited to be applicable in practical environments. The authors therefore extended the SBP so that non-standard features such as release dates, due dates, assembly structures, split structures, overlapping operations, setup times, transportation times, parallel machines and in-process inventory can be modelled (see Ivens and Lambrecht [13]). Recent extensions include the use of resource calendars and the possibility that operations require more than one resource at a time. In addition, other performance criteria could be considered. A brief overview of some extensions is given below in section 3.3.3. We will first discuss the disjunctive graph representation of the scheduling problem and next we will explain the approach of the Extended Shifting Bottleneck Procedure (ESBP) to solve the problem.

3.3.1 The Network Representation of the Job Scheduling Problem

Problems solved by the Shifting Bottleneck Procedure are represented by an activity-on-the-node disjunctive graph $DG = (N, A, E)$. $N = N' \cup \{b\} \cup \{e\}$ with N' the set of nodes each representing one operation, $\{b\}$ and $\{e\}$ are dummy nodes indicating the start and the end of the schedule. Each node $i \in N'$ has a label p_i, the processing time of operation i. A is the set of directed (conjunctive) arcs, representing precedence relations between nodes (i, j) of N. E is the set of undirected (disjunctive) arcs, which represent precedence relations between nodes (i, j) of N' that require the same resource. Initially, the arcs E are undirected. A solution to the problem corresponds with the choice of a direction for each arc in E. Let E' be such a selection, i.e. a set of directed arcs. $DG' = (N, A, E')$ is the graph obtained when replacing the disjunctive arcs in DG by the selection E'. The longest path in DG' from $\{b\}$ to $\{e\}$ corresponds to the makespan of the schedule. A feasible solution requires that DG' is acyclic. The set $E(E')$ can be partitioned into m subsets $E_1, \ldots, E_m$ $(E'_1, \ldots, E'_m)$ where each set E_k represents the disjunctive arcs connecting nodes which require the same resource k. The job shop scheduling problem consists of finding a selection which minimizes the longest path between the starting and ending node.

3.3.2 The Shifting Bottleneck Procedure

The Shifting Bottleneck Procedure is an iterative procedure which schedules one resource at a time. In each iteration two decisions have to be taken, namely we have to decide which resource is the bottleneck and we have to determine the sequence on that resource. Next, at the end of each iteration there is a reoptimization run on all resources scheduled so far. To formalize this, let M_0 be the set of all scheduled resources. The Shifting Bottleneck Procedure logic can be summarized as follows:

- Step 0 $M_0 = \{\}$
- Step 1 Identify the bottleneck resource m among resources $M \setminus M_0$, and calculate its optimal sequence, given the partial schedule on resources M_0. $M_0 = M_0 + \{m\}$
- Step 2 Reoptimize successively the sequence of each resource $m \in M_0$, given the partial schedule on resources $M_0 \setminus \{m\}$. If $M_0 = M$ then stop, else go to step 1.

Both for identification and scheduling of the bottlenecks, as well as for the reoptimization, relaxations of the original problem are used. Problem $P(M_0, m)$ is the single machine relaxation for resource m of the original job shop scheduling problem, taking into consideration all selections already made on scheduled resources. With each resource $j \in M_0$, a selection E'_j is associated, describing the directions for disjunctive arcs of already scheduled resources. Let $DG'_m(M_0)$ be the graph obtained from DG when replacing all the disjunctive arcs of the resources in M_0 by their selections and by deleting all remaining disjunctive arcs except these of resource m. Let r_i be the length of the longest path in $DG'_m(M_0)$ from node $\{b\}$ to node i, then r_i is the earliest starting time of operation i, given the partial schedule determined by resources in M_0 and given a relaxation of resources other than m. For this reason, r_i is called the head of operation i. Similarly, there is q_i, the tail of operation i which equals the length of the longest path from node i to the end node $\{e\}$. In this way, q_i is the amount of work that has to be performed after operation i and before the schedule can end. For the standard job shop scheduling problem, heads and tails can be calculated within linear time complexity (see Adams, Balas and Zawack [1]). Consider now the problem $P(M_0, m)$:

$$\begin{array}{lll} \min & C & \\ \text{Subject to} & C \geq t_i + p_i + q_i & \forall i \in N_m \\ & t_i \geq t_j + p_j \vee t_j \geq t_i + p_i & \forall (i,j) \in E_m \\ & t_i \geq r_i & \forall i \in N_m \end{array}$$

where N_m is the set of all operations which require resource m for processing. Problem $P(M_0, m)$ is equivalent to a single machine scheduling problem with release dates and due dates. Although it is a NP-hard problem, relatively large instances can be solved by a branch-and-bound procedure proposed by Carlier [10]. For most practical applications, computer time needed to solve $P(M_0, m)$ constitutes no problem. The solution of $P(M_0, m)$ can be seen as a measure of the 'constraintness' of resource m, given that the resources in M_0 are already scheduled and given that the constraints on unscheduled resources other than m are relaxed. The larger the objective function of this problem, the more likely that resource m is a bottleneck. To identify the bottleneck among unscheduled resources, the resource with the worst objective function is selected (i.e. the resource m^* for which $P(M_0, m^*)$ is the minimum over all $m \in M_0$ of $P(M_0, m)$; the sequence on this new bottleneck is the sequence found by solving $P(M_0, m^*)$).

The reoptimization step goes as follows. Let $R_1, \ldots, R_{|M_0|}$ be an ordering of the resources in M_0 (e.g. the order in which they were scheduled). Then, for $m = 1, \ldots, |M_0|$ solve $P(M_0 \setminus \{R_m\}, R_m)$ and replace E'_{R_m} by the new sequence found. This reoptimization cycle can be repeated a number of times. Note that in the reoptimization heads and tails need to be recalculated each time that a different resource R_m is considered.

3.3.3 The Extended Shifting Bottleneck Procedure

In this section we will briefly describe the ESBP which is more suitable for real-life applications. For an in-depth treatment of these extensions we refer to Ivens and Lambrecht [13]. The extensions are modeled by an Extended Disjunctive Graph (EDG). This representation is similar to the DG, but arcs can have labels to represent general precedence relationships such as Start-Start (SS), Start-Finish (SF), Finish-Start (FS) and Finish-Finish (FF). In practice, it is common to allow overlapping, i.e. units of a batch do not have to wait till the whole batch is finished, instead, products flow from one resource to another in several smaller transfer batches. When preemption of processing is not allowed, overlapping can be modelled by using SS precedence relations (or negative FS). Overlapping may have a considerable impact on lead time performance. Also all sorts of delays, forced waiting times or cooling times can be modeled by FS precedence relationships.

In the standard job shop problem, operations can have only one technological predecessor or successor. Product assemblies and splits can easily be modelled by allowing multiple predecessors and successors.

Customer or manufacturing lots can have a release date and a due date. These can be incorporated in the EDG by assigning FS precedence relations respectively between {b} and the first operation(s) of the lot (in case of release dates) or between the last operation(s) of the lot and {e}. We also allow restrictions on starting times or finishing times of individual operations (e.g. due to temporary unavailability of raw materials).

In the standard job shop scheduling problem there is only one unit available of each resource. For some applications however, it is possible that some resources are available in multiple units (i.e. parallel machines). Thus, in addition to sequencing, an assignment of operations to resources has to be done. The EDG is able to model this complication. In each iteration of the shifting bottleneck procedure a resource type is scheduled, which requires the solution to a single or parallel machine problem, depending on the availability of the resource.

3.3.4 The Metal Shop Example

We will now illustrate the Extended Shifting Bottleneck Procedure with the small metal shop example. The eight manufacturing orders from table 4 have to be scheduled. The corresponding disjunctive graph is given in figure 3. Each node corresponds with one operation. The numbers above the nodes indicate the processing time of the operation. In order to keep the figure transparent, the operations which have to be scheduled on the same machine share the same shade.

Currently, two earlier released manufacturing orders reside on the shop floor. A first manufacturing lot for 5 units of product P has already passed the cutter and has been loaded on the grinder, on which the setup (20 hours work content) is already performed. So we still have to schedule 50 hours processing time for the operation on the grinder and a complete operation on the lathe (84 hours). A second manufacturing order for 6 units of product S has already spend 34 hours on the lathe. In addition to these two in-process orders, there are three manufacturing orders with three operations and five manufacturing orders with two

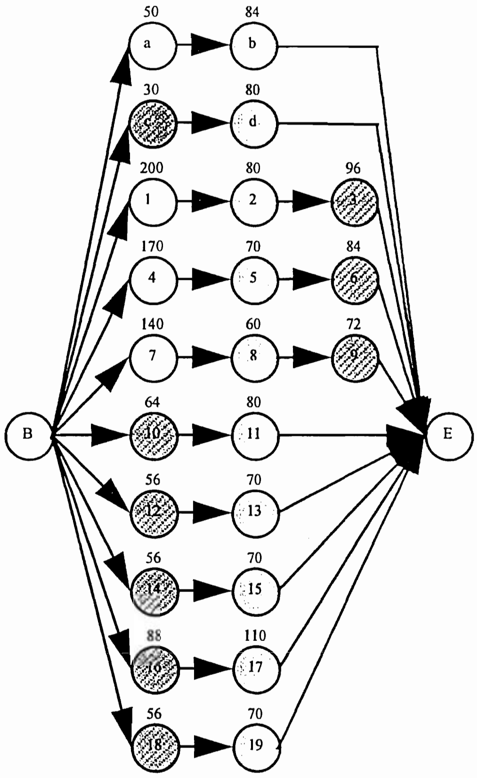

Figure 3: The disjunctive graph of the small metal shop

operations. For instance nodes 4,5 and 6 symbolize manufacturing lot L_{P2}. Node 4 has a processing time of 170 hours (20 hours setup and 5*30 hours processing).

The disjunctive graph contains all the necessary input for the ESBP. The output of the ESBP is visualized in figure 4 as a gantt-chart. The gantt-chart shows the two manufacturing orders already in process and the eight newly released manufacturing orders. In this small example, for all operations scheduled, slack time is available as explained earlier.

3.3.5 The Execution Phase

In this phase, the detailed schedule will be executed. From the detailed schedule, dispatching and picking lists can be drawn. A data captation system can transmit information concerning work progression back to CLIPS. From time to time, a recalculation of the detailed schedule will be necessary because of the numerous changes on the shop floor. The frequency of recalculation is of course a function of the dynamics of the shop. A reoptimization of the lot sizing and lead time estimation phase will also be required now and then but of course less frequently.

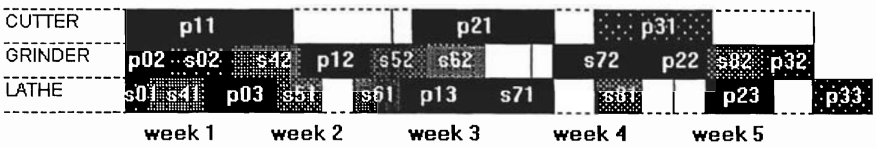

Figure 4: The gantt chart of the detailed schedule

4 Real life application

The methodology outlined in this paper is well suited for real life applications. In this section we report on an implementation of CLIPS in a medium sized metal working company. We will stress the applicability of our approach and focus on the resulting benefits and savings. The moderate computational effort required for solving the various modules of CLIPS and the high quality solutions obtained, turn the procedure into an ideal decision support system for achieving continuous improvement.

The metal working company we considered produces transmissions for off-road vehicles. There are four major processing steps: first a job shop where unhardened components (shafts and gears) are produced out of raw steel components; then a furnace operation is performed on the shafts and gears to harden these components; third, the components are finished through a number of grinding operations; finally, they are assembled into castings on an assembly line. Our application focusses on the first shop where the raw steel components are transformed into shafts and gears. The shop orders stem directly from the MRP requirements demanded by the furnace. In the future CLIPS will be expanded to all shops, making the current MRP system of the company superfluous. The subsystem we consider consists of 70 machines and produces 550 different components which results in 3,000 different operations. On a yearly basis, this metal shop handles about 10,000 customer orders.

We used the requirements of both (actual) customer orders and forecasts over a one year time horizon, to estimate the arrival patterns of manufacturing orders. Feeding the current MRP lot sizes (heuristically fixed at 1, 2, 4, 6, 8, 12 or 16 weeks of supply) of the company into the queueing approximation results in an average lead time per operation of 68 hours. Then, we optimized the lot sizes of the 550 products on a Pentium 60 Mhz PC. After 7.5 hours, our dedicated optimization routine obtained a global optimum of 22 hours lead time per operation. The new lot sizes reduce the original lead times by 67.7 %. The computational effort for the optimization is rather high, but we forced an accuracy of 10^{-10}, which is not neccessary for practical applications. Moreover, the optimization routine is run infrequently. An evaluation of a given set of lot sizes takes only about 0.3 seconds, which allows the practitioner to evaluate what-if data modifications quickly.

Next, we examined the resulting lot sizes and the resulting lead times in the tuning phase. Here, often unsatisfactory lead times are detected. To remedy this, the various options of the tuning phase can be used. For instance, extremely large lead times occur at heavy loaded machines. Off-loading and the provision of additional capacity (extra shifts or overtime) can cause a considerable decrease in utilization and consequently in the average and variance of the lead times. It is crucial that all serious capacity problems, whether permanent of occasionally, are solved before diving into the detailed scheduling phase. A detailed analysis revealed that there was one machine with a utilization level of 97 %, resulting in unsatisfactory long average waiting times of 226 hours on that machine. We increased the capacity of that machine with one shift and we ran the lot sizing and lead time estimation routine again. This resulted in a lead time of 16.9 hours per operation. This is an additional improvement of 23 % achieved by adjusting the capacity of only one resource.

In the scheduling phase, we group the various customer orders and forecasts (for a one year time horizon) into new manufacturing orders with the aid of a dynamic program. After determining a release date and a due date for each manufacturing order (with a service rate of 95%), we run the Extended Shifting Bottleneck Procedure. To test our approach we released all new lots resulting in a huge detailed scheduling problem of about 30,000 operations. For the practical application however, the scheduling problems are much smaller since it is not necessary to schedule all manufacturing orders, we can restrict ourselves to manufacturing orders whose release date is within an acceptable planning period. In table 6 we compare the outcome of our detailed schedule with current practice in the company. In the first column we evaluate the schedule methods used on the floor before the introduction of CLIPS. These results include the old MRP lot sizes, no intelligent way of grouping customer orders and a slack based priority scheduling rule. In the second column we display the CLIPS outcome.

performance measure	without CLIPS	with CLIPS
proportion of late orders	13.20 %	10.63 %
maximum lateness	19.57 days	3.71 days
average lateness	-29.61 days	-5.00 days
average tardiness	0.60 days	0.14 days
average leadtime	15.72 days	2.31 days

Table 6: The overall performance of the detailed schedule

On all performance measures CLIPS outperfomed the traditional planning approach. The maximum lateness is drastically reduced. The average (in process) lead time decreased by 85%. The average lateness increases to -5 days, which indicates that the lead time estimation is much more accurate, resulting in enormous lead time and work in process savings. Still, 10.63% of the orders is late. This is of course a major concern. The main reason is that in certain time periods the shop experiences a heavy workload, too many orders are due in a small time period. In order to make comparison with current practice meaningful we did not manipulate release dates, due dates or capacity availability. However the tuning phase offers plenty of opportunities to manage the workload better, so that late orders can be avoided. The distribution of the lateness however is seriously improved with the CLIPS approach.

In order to fully obtain the benefits of the proposed hierarchical approach suggested in this paper one has to focus on data accuracy and on some behavioral aspects of scheduling. As mentioned above, the current practice is based on priority rules. This myopic approach has to be replaced by a scheduler looking at all machines simultaneously. This results in a dispatch sequence which is not always preferred by the operators who are used to set priorities autonomously. In order to overcome this, one has to spend a lot of time on the floor to introduce this overall look and to make sure everybody is confident with the proposed priorities.

5 Conclusion

In this paper we proposed a general methodology to analyze and schedule a job shop. A four phase methodology is proposed including a lot sizing and lead time estimation phase, a tuning phase, a scheduling phase and an execution phase. In each phase we use analytic approaches which are suitable for real life applications. The methodology is illustrated with an example and a real life application is given. The CLIPS methodology is embedded in a software package. Our practical experience indicates that our approach has a great potential both in terms of computational effort required and in terms of the quality of the generated schedules.

A Appendix

The variance of the batch waiting time is approximated by

$$\begin{aligned}
V(Wq_m) &= [E(Wq_m)]^2 c^2_{Wq_m} \\
c^2_{Wq_m} &= \frac{c^2_{Dq_m}+1-\sigma_{q_m}}{\sigma_{q_m}} \\
\sigma_{q_m} &= \rho'_m + (c^2_{ba_m}-1)\rho'_m(1-\rho'_m)h(\rho'_m, c^2_{ba_m}, c^2_{bs_m}) \\
h(\rho'_m, c^2_{ba_m}, c^2_{bs_m}) &= \begin{cases} \frac{1+c^2_{ba_m}+\rho'_m c^2_{bs_m}}{1+\rho'_m(c^2_{bs_m}-1)+\rho'^2_m(4c^2_{ba_m}+c^2_{bs_m})} & c^2_{ba_m} \le 1 \\ \frac{4\rho'_m}{c^2_{ba_m}+\rho'^2_m(4c^2_{ba_m}+c^2_{bs_m})} & c^2_{ba_m} \ge 1 \end{cases} \\
c^2_{Dq_m} &= 2\rho'_m - 1 + \frac{4(1-\rho'_m)d^3_{bs_m}}{3(c^2_{bs_m}+1)^2} \\
d^3_{bs_m} &= \begin{cases} \frac{3}{4}\left[\frac{1}{q^2_m}+\frac{1}{(1-q_m)^2}\right] & c^2_{bs_m} \ge 1 \\ (2c^2_{bs_m}+1)(c^2_{bs_m}+1) & c^2_{bs_m} < 1 \end{cases} \\
q_m &= \frac{1}{2} + \sqrt{\frac{c^2_{bs_m}-1}{c^2_{bs_m}+1}}
\end{aligned}$$

where $c^2_{Wq_m}$, the scv of the batch waiting time, σ_{q_m}, the batch probability of delay, $P(Wq_m > 0)$, $c^2_{Dq_m}$, scv of the conditional batch waiting time i.e. the batch waiting time, given that the server is busy and $d^3_{bs_m} = E[S^3_{b_m}]/E(S_{b_m})^3$.

References

[1] J. Adams, E. Balas, and D. Zawack. The shifting bottleneck procedure for job-shop scheduling. *Management Science*, 34(3):391–401, 1988.

[2] S. L. Albin. *Approximating queues with superposition arrival processes.* PhD thesis, Department for Industrial Engineering and Operations Research, Columbia University, 1981.

[3] S. L. Albin. Approximating a point process by a renewal process, II: superposition arrival processes to queues. Technical report, Department of industrial engineering, Rutgers University, 1982.

[4] D. Applegate and W. Cook. A computational study of the job-shop scheduling problem. *ORSA Journal of Computing*, 3(2):149–156, 1991.

[5] K. R. Baker. Requirements planning. In S.C. Graves, A.H.G. Rinnooy Kan, and P.H. Zipkin, editors, *Logistics of production and inventory*, chapter 11, pages 571–627. North Holland, 1993.

[6] E. Balas. On the facial structure of scheduling polehedra. *Mathematical Programming Study*, (24):179–218, 1985.

[7] E. Balas, J.K. Lenstra, and A. Vazacopoulos. The one-machine problem with delayed precedence constraints and its use in job shop scheduling. *Management Science*, 41(1):94–109, 1995.

[8] P. Brucker, B. Jurisch, and A. Krämer. The job shop problem and immediate selection. Technical report, Univ. Osnabrück / Fachbereich Mathematik, 1992.

[9] P. Brucker, B. Jurisch, and B. Sievers. A fast branch-and-bound algorithm for the job-shop scheduling problem. Technical report, Univ. Osnabrück / Fachbereich Mathematik, 1991.

[10] J. Carlier. The one-machine scheduling problem. *European Journal of Operational Research*, 11:42–47, 1982.

[11] J. Carlier and E. Pinson. An algorithm for solving the job-shop problem. *Management Science*, 35(2):164–176, 1989.

[12] S. Dauzère-Péres and J.B. Lasserre. A modified shifting bottleneck procedure for job-shop scheduling. *International Journal of Production Research*, 31(4):923–932, 1993.

[13] P. L. Ivens and M. R. Lambrecht. Extending the shifting bottleneck procedure to real-life applications. *European Journal of Operational Research*, To appear.

[14] U. S. Karmarkar. Lot sizes, lead times and in-process inventories. *Management Science*, 33(3):409–423, 1987.

[15] B.J. Lageweg, J.K. Lenstra, and A.H.G. Rinnooy Kan. Job shop scheduling by implicit enumeration. *Management Science*, 24:441–482, 1977.

[16] M. R. Lambrecht, P. Ivens, and N. J. Vandaele. Lead time minimization in a job shop environment. Onderzoeksrapport To appear, Departement Toegepaste Economische Wetenschappen, Katholieke Universiteit Leuven, 1995.

[17] M. R. Lambrecht and N. J. Vandaele. A general approximation for the single product lot sizing model with queueing delays. *European Journal of Operational Research*, To appear.

[18] Y. Leung and R. Suri. Performance evaluation of discrete manufacturing systems. *IEEE Control Systems Magazine*, 10:77–86, 1990.

[19] J.G. Shantikumar and J.A. Buzacott. Open queueing network models of dynamic job shops. *International Journal of Production Research*, 19(3):255–266, 1981.

[20] L. M. Wein and P. B. Chevalier. A broader view of the job shop scheduling problem. *Management Science*, 38:1018–1033, 1992.