OR Spectrum (2013) 35:251–290
DOI 10.1007/s00291-011-0267-y

REGULAR ARTICLE

Lead time anticipation in Supply Chain Operations Planning

Michiel M. Jansen · Ton G. de Kok · Jan C. Fransoo

Published online: 15 July 2011

Abstract Linear programming (LP) models for Supply Chain Operations Planning are widely used in Advanced Planning Systems. The solution to the LP model is a proposal for order releases to the various production units (PU) in the supply network. There is a non-linear relationship between the work-in-process in the PU and the lead time that is difficult to capture in the LP model formulation. We propose a two-step lead time anticipation (LTA) procedure where the LP model is first solved irrespective of the available production capacity and is subsequently updated with aggregate order release targets. The order release targets are generated by a local smoothing algorithm that accounts for the evolution of the stochastic workload in the PU. A solution that is both feasible with respect to the planned lead time and meets the material requirements may not exist. By means of discrete event simulation, we compare a conservative strategy where the production quantities are reduced to an optimistic strategy where the planned lead time constraint is allowed to be violated.

Keywords Supply chain management · Hierarchical production planning · Lead time anticipation · Production smoothing

1 Introduction

Advances in information and communication technology have provided firms with access to detailed and up-to-date information about the state of their primary process.

We acknowledge two anonymous referees for their comments which helped to improve this paper considerably.

M. M. Jansen (✉) · T. G. de Kok · J. C. Fransoo
School of Industrial Engineering, Eindhoven University of Technology,
P.O. Box 513, Paviljoen E.14, 5600 MB Eindhoven, The Netherlands
e-mail: m.jansen@tue.nl

This enables them to exercise centralized control of their large and complex supply networks. Information sharing across partners in the supply chain, access to point-of-sales data, and integrated sales-and-operations planning, provide valuable advance demand information. On the other hand, a volatile and demanding market requires firms to be flexible and provide a reliable supply of a broad range of products, preferably off-the-shelf. Firms are therefore looking for ways to reduce lead times and inventories to gain competitive advantage while being encumbered by smaller sales quantities and shorter product life cycles for individual products. As a result, the problem of coordinating material flows and release of production orders has become much more difficult. Decision making is particularly difficult due to many forms of uncertainty in the information upon which decision are based. Different forms of slack (capacity, time, and inventory) are created in order to deal with these uncertainties. In this environment, there is a need for reliable dynamic planning methods that efficiently utilize available information on the one hand while taking into account the uncertainties that are inherent to production and demand processes. The problem is complex and optimal policies are intractable. This has led to the use of linear programming (LP) models applied in a rolling schedule based fashion. The stage for these methods are Advanced Planning Systems (APS).

LP models are not well capable of capturing the non-linear relationships between work-in-process (WIP) and lead time. In this paper we present a method that augments generic LP models for Supply Chain Operations Planning (see for example Spitter et al. 2004; De Kok and Fransoo 2003) with the ability to account for uncertainties in the production and demand processes. We do so by separating the coordination of the goods flow from the loading decisions for the production unit. This separation is facilitated through the planned lead time concept. Before we describe the approach in detail, we discuss the hierarchical setting of this paper and related literature. Using discrete-event simulation, we evaluate the performance of our method in comparison with other approaches.

It is the objective of Supply Chain Operations Planning (SCOP) to "coordinate the release of materials and resources in the supply network under consideration such that customer service constraints are met at minimal cost" (cf. De Kok and Fransoo 2003). The supply network under consideration is a network of production units (PU) that are separated by stock points. A PU deploys its resources to carry out production activities that transform one or more input materials into output materials. Coordination of these activities is required because both material availability and resource capacity is finite.

A PU may be a single machine, a production line, or a whole department. In general, we assume that the supply network belongs to a single organization. If parts of the production are outsourced, we assume that dedicated capacity is made available to the supply network. We assume that information about inventories and WIP is shared freely in the supply network and that there is central coordination of order releases.

SCOP should be considered as part of a hierarchical production planning framework. The framework for this paper is derived from the work of Bertrand et al. (1990) and is shown in Fig. 1. We believe that this simple framework reflects the structure of (operational) planning in practice. Other, more comprehensive frameworks for hierarchical production planning are found in Vollmann et al. (1984), Bitran and Tirupati (1993), Meyr et al. (2005) and Hopp and Spearman (2001). We refer to

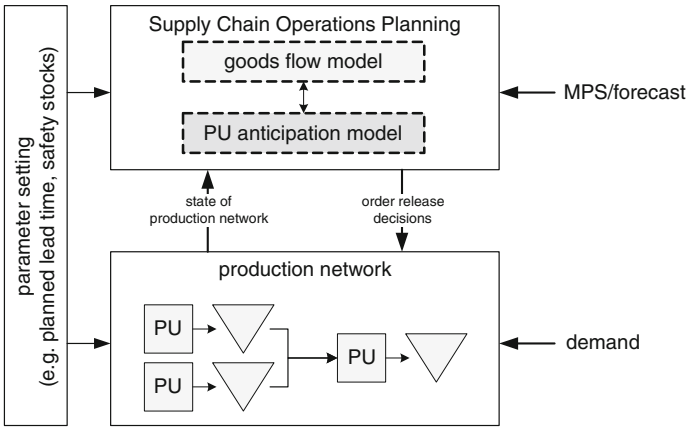

Fig. 1 Framework

De Kok and Fransoo (2003) for a detailed discussion of the hierarchical planning concept. A formal treatment of hierarchical decision making can be found in Schneeweiss (2003).

The SCOP function forms the top level in the framework. It takes as input the state of the supply network (inventory levels) and a demand forecast and sends instructions to the PUs in the form of order release decisions. The SCOP function is configured by parameters such as planned lead times and safety stocks. The SCOP model of the supply network consists of two aspects. The first aspect is the specification of the consumption of materials (the goods flow model) and the second aspect is the specification of the consumption of capacity of (non-inventoriable) resources (the anticipation function).

The PU control functions form the lower level in the framework. Note there is no control of goods flows between PUs at this lower level so we may consider the PU individually. We think of a PU as a complex entity with an autonomous scheduling or dispatching function. It has its own control paradigm and objectives which are partly configured by the same parameters as the SCOP function. The PU control function takes as input the order release decisions of the SCOP level.

The *anticipation function* is the representation of PU dynamics in the SCOP model. The anticipation function is necessarily an abstraction of the actual dynamics in the PU for three reasons. First, the PU has a degree of freedom to optimize its own objectives, independently of the SCOP function. Second, not all events that take place in the PU are known a priori or can be captured in mathematical formulations. Finally, a detailed description of the dynamics is often not computationally tractable.

The purpose of an anticipation function is to generate schedules of order releases that lead to efficient use of available capacity, thereby reducing the need for slack in the supply network. Typically, slack takes the form of inventories that act as a buffer for temporary deviations between supply and demand. Such "safety stocks" are fixed in the mid-term. Especially if there is foreknowledge of the requirements, high safety stock requirements can be avoided by anticipating expected mismatches between the available production capacity and the demand by producing ahead of requirements.

The result is a "smoothed" schedule of planned order releases. The smoothed schedule effectively yields buffer stocks as well, but contrary to safety stocks these buffers only exist for a short period of time.

In this paper we consider production processes where processing times are stochastic. Such production processes occur naturally where people are involved or where environmental influences play a role. Stochasticity also is the consequence of the hierarchical structure of decision making. Many decisions at the PU level cannot be controlled or predicted at the SCOP level. We note that uncertainty may take many different forms which are not always directly reflected in the individual processing times. This is, for example, the case for unplanned machine downtime. However, stochastic processing times can act as a proxy for various forms of uncertainty. The substitution of various forms of uncertainty into stochastic processing times is also referred to as the principle of *effective processing times* (cf. Hopp and Spearman 2001).

It is well known that stochasticity in processing times affects the congestion in the PU. An anticipation function that fails to take into account the effect of stochasticity on the congestion in the PU, is likely to underestimate the lead time of a production order. An important class of anticipation functions that explicitly account for congestion effects caused by stochastic processing times is the class of *clearing functions*. Clearing functions relate the output of a PU to the work-in-progress (WIP) at the start of a period. The clearing function has a concave shape that depends on the variance of the processing times and approaches the nominal capacity asymptotically as the WIP increases. The clearing function is a very powerful tool since it can be approximated by linear constraints that are directly inserted in LP formulations of the SCOP model. However, the clearing function has two important drawbacks.

First, the clearing function requires specification of the order of processing in a multi-item setting. This topic is extensively discussed in Asmundsson et al. (2009). The specification of the order of processing is undesirable because it restricts the decision space of the PU. It is shown in Selçuk (2007) that the performance of the PU is reduced if detailed instructions on the order of processing are given.

The second drawback of the clearing function is that its shape is fixed by a parameter whereas in fact the optimal parameter depends on the variance of the workload. This fact is also recognized in Missbauer (2010) where it is proposed that the initial workload in a period has some distribution that depends only on the mean of the workload. However, also the variance of the workload is a function of the order release decisions and typically changes over time.

The observations in the previous two paragraphs bring us to the first two contributions in this paper:

1. We propose an alternative anticipation function that focuses on throughput during the lead time rather than periodic throughput, thereby avoiding the need to make assumptions about the order of processing in a multi-item setting.
2. We propose a solution procedure in which not only expectation but also variance of the workload is tracked over the horizon.

Besides smoothing the schedule of planned order releases, the anticipation function also facilitates the detection of capacity infeasibility. A capacity infeasibility occurs if, over a number of periods starting from the current period, cumulative expected

requirements exceed cumulative expected capacity. If there is a perfect deterministic anticipation function, then it is reasonable not to release more work to the PU than the amount that is anticipated to be produced since releasing more work does not lead to a higher output. In a setting where processing times are uncertain, the optimal response to a capacity infeasibility is less trivial. A higher workload results in a higher expected output but at the same time increases the probability of tardy orders. Increased numbers of tardy orders in turn lead to a higher safety stock requirement. This leads us to formulate the third contribution in this paper.

3. We compare two strategies for dealing with a capacity infeasible schedule of order releases. In the conservative schedule, reliability of the lead time is preferred over higher output. In the optimistic schedule we prefer the higher expected output.

The remainder of this paper is organized as follows: In Sect. 2 we describe the SCOP and PU model in detail and discuss the most important assumptions. In the same section we also discuss the role of planned lead times as a hierarchical decoupling mechanism. Next, we summarize briefly the clearing function and other anticipation functions found in the literature in Sect. 3. In Sect. 4 we first give an overview of the solution approach that we propose and then discuss the details. In the same section, we also compare the conservative and optimistic strategies for dealing with capacity infeasibility. In Sect. 5 we verify by means of a discrete event simulation experiments whether our approach meets the objective to reduce the total inventory needed to meet customer service levels. Finally, we summarize the conclusions and discuss topics for further research in Sect. 6.

2 The SCOP and PU models

In this section, we describe the SCOP and PU models for this paper and the main assumptions that we make. We introduce the notation and formulate the SCOP and PU model in mathematical terms. We present a standard multi-item multi-resource SCOP model formulation with a deterministic anticipation function. It is the topic of this paper to replace this deterministic anticipation function by one that accounts for stochasticity in the PUs.

The SCOP models that we consider are applied in a rolling schedule context (cf. Simpson 1999). Periodically, a production plan is generated for a number of planning periods in the future. This number is called the planning horizon. Planning periods may be days, weeks, or months, depending on the desired release frequency. Only the decisions for the first planning period are implemented. At the start of the next planning period, a new production plan is generated using up-to-date forecasts and up-to-date information about the state of the supply network.

Furthermore we consider SCOP models that can be formulated as a linear program. At present, this seems to be the only way to generate a coordinated schedule of planned order releases for general, multi-item, multi-echelon networks with finite capacities. Besides, these are the type of model formulations found in today's Advanced Planning Systems (Stadtler 2005).

We make the following assumptions about the supply network:

1. *Multi-item PU* A PU can process multiple items and each item can be processed in a single PU only.
2. *Feedforward networks* The supply network is a feedforward network. That is, any two PUs can be ordered such that, on all routes through the supply network that pass through both PUs, one PU is always visited before the other.
3. *Planned Lead Times* We assume that there is a single planned lead time per PU. The planned lead time is an input parameter to both the SCOP level and the PU control functions and we assume that the PU will give preemptive priority to goods that are released at an earlier date if this is necessary to meet the planned lead time. We note here that our approach can be extended to item dependent planned lead times.
4. *Make-to-stock* We assume products are produced to stock. However, the methods presented in this paper can easily be extended to the control of supply chains whose final stages operate on a make-to-order bases.
5. *Relevant costs* The objective of the SCOP function is to minimize the opportunity costs of order releases. In this paper we consider the costs of holding inventory only. We distinguish between costs for holding work-in-progress (WIP) and costs for holding items in finished goods inventory (FGI). We assume that the WIP and FGI costs for items are proportional with a common ratio f^{wh}.
6. *Predefined transport batch-sizes* We do not consider lot sizing decisions in this paper. We assume that the unit of measurement for release, production, and demand quantities is a transport batch. The processing time for a unit of production is the effective processing time as defined in chapter 8 of Hopp and Spearman (2001).
7. *Demand process* There is a stochastic dynamic demand for end items and there exists an unbiased forecast of demand.
8. *Non-stockout service level* The planning system adheres to a non-stockout service level constraint ψ, also referred to as the P1 service level (Silver et al. 1998). Safety stocks of end-items ensure that the service level is satisfied.

2.1 Notation

The following is a list of notations used in this paper. The planning horizon is H and planning periods are indexed by $t = 0, \ldots, H-1$. Without loss of generality, we assume that the current period is $t = 0$. PUs are indexed by $k = 1, 2, \ldots$ and items are indexed by $i = 1, 2, \ldots$. The set of all items is denoted by U. The PU that processes item i is denoted by $\kappa(i)$ and the set of all items produced in PU k is denoted by U_k.U_E is the set of all items that face external demand. The available processing time in PU k is C_k and the processing time for a single unit of item i has expectation μ_i and variance σ_i^2. The bill-of-material is specified by $\{a_{ij}\}_{i,j \in U}$ where a_{ij} specifies the quantity of item i consumed by producing one unit of item j. We define $U_S(i) := \{j : a_{ij} > 0\}$ to be the set of all successors of item i. The planned lead time for a PU is denoted by L_k and specifies the number of planning periods that are allowed to the PU to process a production order. Demand of end-item i in period t is denoted by $D_i(t)$ and the forecast is equal to expected demand which is denoted by $\hat{D}_i(t)$. The long-run

average demand in a period for item i is denoted by $\lambda_i := \lim_{t\to\infty} \frac{\sum_{s=0}^{t} D_i(s)}{t}$. The long-run resource utilization rate for PU k is defined as $\rho_k := \frac{\sum_{i\in U_k} \lambda_i \mu_i}{C_k}$.

The decision variable for the planning algorithm is the planned order release quantity for an item i in period $t = 0$ denoted by $R_i(0) = \hat{R}_i(0)$. For $t > 0$, planned order releases are denoted by $\hat{R}_i(t)$. Due to the rolling schedule principle, this quantity may differ from the actual releases $R_i(t)$ and we denote the difference by

$$\epsilon_i(t) = R_i(t) - \hat{R}_i(t)$$

For $t < 0$, $R_i(t) = \hat{R}_i(t)$ is understood to be the quantity of item i released at a previous time, $-t$ periods before the current period.

For each end-item $i \in U_E$ there may be a safety stock denoted by SS_i. The surplus and deficit of inventory over the safety stock are denoted, respectively, by $I_i^+(t)$ and $I_i^-(t)$ so that the net stock level of item i is $SS_i + I_i^+(t) - I_i^-(t)$. We assume that $SS_i = 0$ for all $i \notin U_E$. We make this assumption as optimization of safety stocks under rolling schedule concepts is to-date an unsolvable and intractable problem. Furthermore, earlier experiments in De Kok and Fransoo (2003) suggest that setting $SS_i = 0$ for $i \notin U_E$ is appropriate in situations where value added downstream in supply networks is relatively small. This holds for many practical situations. We also note that our objective is to compare different SCOP concepts under comparable circumstances.

The cost for holding a unit of positive net stock for item i is c_i^h per period. The levels of $I_i^+(t)$ and $I_i^-(t)$ are measured at the start of period t just before receipts of orders released at the start of period $t - L_{\kappa(i)}$. Their planned values are denoted by $\hat{I}_i^+(t)$ and $\hat{I}_i^-(t)$. The quantity of item i produced in period t is denoted by $X_i(t)$ with planned value $\hat{X}_i(t)$.$W_i(t)$ with planned value $\hat{W}_i(t)$ is the work-in-progress (WIP) of item i at the start of period t, just after order releases. The initial WIP at time $t = 0$, just before releases is denoted by $W_i(0-)$. The cost for holding a unit of WIP for item i is c_i^w per period. This cost is proportional to the cost of carrying FGI. That is $c_i^w = f^{wh} c_i^h$.

We also define a number of aggregate variables. The aggregate output of PU k in period t is

$$Y_k(t) = \sum_{i\in U_k} \sum_{n=1}^{X_i(t)} v_{i,n}, \tag{1}$$

the aggregate workload in PU k at the start of period t is

$$V_k(t) = \sum_{i\in U_k} \sum_{n=1}^{W_i(t)} v_{i,n}, \tag{2}$$

and the aggregate releases to PU k at the start of period t are

$$B_k(t) = \sum_{i\in U_k} \sum_{n=W_i(t)-R_i(t)}^{W_i(t)} v_{i,n}, \tag{3}$$

Table 1 Overview of notation

H	Planning horizon
U	Set of all items
U_k	Set of all items produced on PU k
U_E	Set of all planning end-items
$U_S(i)$	Set of all successors of planning item i
a_{ij}	Number of items of type i consumed when producing one item of type j
$\kappa(i)$	PU in which item i is produced
L_k	Planned lead time for PU k
$D_i(t)$ ($\hat{D}_i(t)$)	Demand (forecasted) for item i in period t
λ_i	Long run expected demand of item i in a period
$I_i^+(t)$ ($\hat{I}_i^+(t)$)	Positive part of net stock (planned) minus target stock of item i at the start of period t
$I_i^-(t)$ ($\hat{I}_i^-(t)$)	Negative part of net stock (planned) minus target stock of item i at the start of period t
$R_i(t)$ ($\hat{R}_i(t)$)	Release quantity (planned) of item i at the start of period t
c_i^h	FGI holding cost per period for item i
c_i^w	WIP cost per period for item i
c_i^b	Backordering penalty cost per period for items i
μ_i	Expected processing time for a unit of item i
σ_i^2	Variance of the processing time for a unit of item i
$W_i(t)$ ($\hat{W}_i(t)$)	Work-in-progress (planned) of item i in period t
$W_i(0-)$	Work-in-progress of item i just before order releases at time $t = 0$
$X_i(t)$ ($\hat{X}_i(t)$)	Quantity (planned) produced of item i in period t
C_k	Amount of time available for processing in PU k in period t
ρ_k	Long run utilization rate for PU k
$Y_k(t)$ ($\hat{Y}_k(t)$)	Aggregate output (expected) of PU k in period t, expressed in amount of processing time
$V_k(t)$ ($\hat{V}_k(t)$)	Aggregate workload (expected) in PU k at the start of period t, expressed in amount of processing time
$V_k(0-)$ ($\hat{V}_k(0-)$)	Current workload (expected) just before releases at time $t = 0$
$B_k(t)$ ($\hat{B}_k(t)$)	Aggregate released work (expected) to PU k at the start of period t, expressed in amount of processing time

where $v_{i,n}$ is the processing time for the nth unit of item i in the WIP at time t. Related to these aggregate variables we define $\hat{Y}_k(t)$, $\hat{V}_k(t)$, and $\hat{B}_k(t)$ to be their expected values.

The variables and parameters used are summarized in Table 1. In addition, we define $(x)^+ := \max\{0, x\}$ and $(x)^- := \max\{0, -x\}$ and the cardinality of a set is denoted by $|.|$ (Table 1).

2.2 The PU model

Although a PU typically encompasses multiple resources, we assume that there is a single bottleneck resource per PU that determines the speed of production and that this resource can be involved in one activity (i.e. processing one job) at a time only. We assume furthermore that all other activities are never critical. That is, we assume that the planned lead time is set such that all other activities can always take place within the planned lead time. The planned lead time provides flexibility to the PU and we assume that the PU uses this flexibility to optimize its own objectives. We therefore assume that the PU delivers items to the downstream stock point no earlier than a planned lead time after their release. We also assume that the PU gives priority to orders released at an earlier time if necessary to meet the planned lead time. Finally, we assume that deliveries to the downstream stock point are made only at the end of a period.

Towards specifying the PU behavior mathematically, we introduce some notation. Since we consider a single PU in this subsection, for brevity we omit the index k here. We define the flow time (s_n) of the nth job to be the time for queueing and processing at the bottleneck resource in isolation, given that it is working in a FCFS, work-conserving manner. Let r_n be the time of arrival (order release) of the nth unit to the PU, and let v_n be the processing time for the nth unit. Finally, let ι_n be the type of the nth job. We assume $\{v_n\}_{n=1}^{\infty}$ is a series of independently distributed random variables and

$$\left.\begin{array}{l}\mathbb{E}[v_n] = \mu_i \\ \mathrm{Var}[v_n] = \sigma_i^2\end{array}\right\} \quad \text{if } \iota_n = i \tag{4}$$

Units are processed in order of arrival. Among the units that arrive at the same time, units are randomly selected for processing. The flow time is given by

$$s_n = (s_{n-1} - (r_n - r_{n-1}))^+ + v_n \tag{5}$$

The PU lead time S_n of the nth job is then given by

$$S_n := \max\{\lceil s_n \rceil, L\} \tag{6}$$

2.3 The SCOP model

The basic SCOP model formulation shown below, is the starting point for this paper and is taken from De Kok and Fransoo (2003). This formulation is not the only formulation possible, but it clearly shows the two aspects of SCOP. The goods flow model is represented in constraints 8 and 9, and the PU anticipation function is represented in constraints (10)–(12). The goods flow (sub)model describes the structure of material consumption and relates release of orders to the availability of goods in time. In the goods flow (sub)model, the requirement for the release of an order is the planned availability of materials. The anticipation function, on the other hand, captures the

consumption of resource capacity. In the anticipation function, the requirement for the release of an order is the planned availability of sufficient capacity to process the order within the planned lead time.

SCOP model formulation (**cf.** De Kok and Fransoo 2003)

$$\text{Min} \quad \sum_{t=1}^{H} \sum_{i \in U} c_i^h \, \hat{I}_i^+(t) + c_i^b \, \hat{I}_i^-(t) \tag{7}$$

S.T

$$\hat{I}_i^+(t+1) - \hat{I}_i^-(t+1) = \hat{I}_i^+(t) - \hat{I}_i^-(t) - \sum_{j \in U} a_{ij} \hat{R}_j(t) - D_i(t) + \hat{R}_i(t - L_{\kappa(i)}), \tag{8}$$

$$\hat{I}_i^-(t+1) - \hat{I}_i^-(t) \leq \hat{D}_i(t), \tag{9}$$

$$\sum_{s=1-L_k}^{t} \hat{R}_i(s) \geq \sum_{s=1-L_k}^{t} \hat{X}_i(s), \tag{10}$$

$$\sum_{s=1-L_k}^{t} \hat{R}_i(s) \leq \sum_{s=1-L_k}^{t+L_k} \hat{X}_i(s), \tag{11}$$

$$\sum_{i \in U_k} \mu_i \, \hat{X}_i(t) \leq C_k, \tag{12}$$

$$\text{for all } i \in U, k = 1, 2, \ldots, t = 0, 1, \ldots, H-1 \tag{13}$$

The objective (7) is to minimize the sum of planned deviations from the target stock SS_i in each period. Positive deviations are denoted by $I_i^+(t)$ and negative deviations are denoted by $I_i^-(t)$. A unit positive deviation is penalized by c_i^h and a unit negative deviation is penalized by c_i^b. Since the planning model assumes that each job resides in the PU for a fixed amount of time (the planned lead time), it is not necessary to include the costs for carrying WIP in the objective function. The parameters c_i^h and c_i^b correspond to FGI holding costs and backorder penalty, but note that they are parameters of the planning model rather than the real (expected) costs. It is generally not possible or necessary to determine real backordering costs. In this paper we assume that the cost for a unit negative deviation of the target is proportional to the cost for a unit positive deviation. Let f^{bh} be the ratio of backordering cost versus holding costs. That is,

$$c_i^b = f^{bh} \, c_i^h \tag{14}$$

Inspired by the equivalence of the optimal order quantity problem with backordering costs and one with a non-stock out probability constraint for a single-item, single-resource model (cf. Zipkin 2000), we set

$$f^{bh} = \frac{\psi}{1 - \psi} \tag{15}$$

The inventory balance constraint (8) links inputs to outputs over time and to the BOM structure a_{ij}. Orders released at the start of period t immediately consume

the material used in production and are planned to be available in the downstream stockpoint in period $t + L_{\kappa(i)}$.

Constraint (9) restricts backordering to external demand only. This constraint makes the optimization model essentially different from the well-known Materials Requirements Planning (MRP) model (Vollmann et al. 1984). In MRP, dependent requirements may result in backorders for components. Planned production in downstream stages depends on the availability of these components so the result is an infeasible solution to the planning problem. The problem thus requires intervention by a human planner who must make allocations of the shortages. The MRP logic as such is therefore an incomplete planning algorithm. In order to compare it to other methods, it would be necessary to formalize the human planner response to infeasibilities which in itself implies the specification of additional logic that completes the planning algorithm. For a more elaborate discussion of this constraint we refer to De Kok and Fransoo (2003). The constraint also implies that backlog penalty costs are irrelevant for intermediate items.

Constraints (10)–(12) form the anticipation function. Constraint (10) specifies that cumulative production does not exceed cumulative releases. Constraint (11) specifies that production orders are processed within the planned lead time. Finally, constraint 12 specifies that the amount of work that can be done in a single period is limited by the capacity. The anticipation function formulated in constraints (10)–(12) ignores the stochasticity of the processing times in the PU. It is the main objective of this paper to replace this deterministic anticipation function by one that accounts for the stochasticity.

2.4 Planned lead times

The planned lead time specifies the time that is allowed to the PU to process a production order. We use the term *planned* lead time to stress that it is an input parameter to the SCOP function and PU control functions. Some authors argue that lead time is endogenous to the SCOP problem since it is a function of the WIP and capacity of the PU. Note that a SCOP model with endogenous lead times follows from the formulation in Sect. 2.3 by setting $L_k = 1$. In that case, Eqs. (10)–(12) reduce to

$$\sum_{i \in U_k} \mu_i \, \hat{R}_i(t) \leq C_k \tag{16}$$

There are several reasons why planned lead times of more than one period may yield better overall planning performance. Planned lead times may need to be longer than one period due to the technical characteristics of the PU. Longer planned lead times also lead to a smoother production plan (see Graves 1986). Most relevant to this paper, however, is the fact that the planned lead time is a means of hierarchical decoupling of decision making. The planned lead time provides a degree of freedom to the PU that allows it to optimize its own objectives. A longer planned lead time also allows for higher levels of WIP which enables the PU to use its resources more efficiently.

It is important to note that the planned lead time is fixed and does not change if production orders are release ahead of time due to production smoothing. If a production

order is rescheduled to be released at an earlier time (in anticipation of a capacity shortage), both the release date and the completion date are changed. The time between the completion of a job and its requirement is an additional slack that exists at the SCOP level only and not in the PU.

3 Literature

In this section we list a number of anticipation functions found in the literature. A well-known framework for modeling production is provided by Hackman and Leachman (1986). A key element of this framework is the *dynamic production function* that maps inputs to outputs for a resource over time. The dynamic production function does not depend on the workload in the PU. However, congestion effects in the PU depend to a largely on the workload. Graves (1986) proposes anticipation function where the periodic throughput is linearly proportional to the workload in the PU but does not explicitly account for congestion.

Clearing functions are anticipation functions that express the throughput in a PU as a function of the workload. Karmarkar (1989, 1993) argues that, if capacity is finite, the clearing function is a concave function that asymptotically grows to the capacity of the PU. Missbauer (2002) distinguishes bottleneck workstations from non-bottleneck workstations. The bottleneck workstations are represented by a queue in steady state, whereas the non-bottleneck workstations are modeled as a first-order exponential delay. For an extensive review of clearing functions, see Pahl et al. (2007).

Asmundsson et al. (2006, 2009) address the problem of disaggregation for multi-item clearing functions. A single-dimensional clearing function only specifies the relation between an aggregate measure of workload and the aggregate expected output. By means of a simple example, they show that it is necessary to make assumptions on how the work-in-progress is actually processed. In their paper, Asmundsson et al. assume that the output for an item will be proportional to its share in the workload. A similar proportionality assumption is made implicitly by Hwang and Uzsoy (2005). They furthermore include the lot size for items in their clearing function.

Most clearing functions presented in the literature are based on the assumption that planning periods are long enough for the system to achieve steady-state conditions. This assumption may not be met in practice. Missbauer (2009) propose a clearing function that is derived from approximations of the transient behavior of a queueing system. The resultant clearing function is not concave and depends on binary variables so it is not straightforward how this clearing function can be included in an LP model. Another transient approach is found in Selçuk (2007) who propose a clearing function that is obtained directly as the expected output conditioned on the workload at the start of the period. This clearing function has a concave shape similar to the steady-state versions, but grows faster to the capacity of the system.

Clearing functions predict the behavior of a PU at a high level of abstraction. Their simple form permits direct inclusion in the LP model. In iterative approaches (Hung and Leachman 1996; Byrne and Bakir 1999; Hung and Hou 2001; Kim and Kim 2001; Riano 2002), the LP model is solved several times where each time the variables that specify the PU behavior are updated. These updated values are obtained

from a separate, descriptive PU model. In Hung and Leachman (1996), Byrne and Bakir (1999) and Kim and Kim (2001), this model is a simulation model and in Hung and Hou (2001) and Riano (2002) this model is the $M/M/1$ queueing model.

In these iterative approaches, release decisions directly depend on the expected state of the system but this state is unknown until the decisions have been evaluated in the anticipation function. The necessity to iterate in these models stems from this circularity. The convergence of the iterative approaches described in the previous paragraph is not at all guaranteed. In Irdem et al. (2008) it is shown that the deviation of objective and parameter values may even diverge in successive iterations. Furthermore, even if convergence is achieved to a satisfactory level, the time required to obtain a solution may be rather long and unpredictable.

4 Lead time anticipation

In this section, we develop an anticipation function that accounts for the stochastic processing times in the PU. We refer to this model as the *lead time anticipation* (LTA) model. The LTA formulations are nonlinear so that they cannot be inserted in the LP formulation of the SCOP model directly. Instead, we present a procedure that iterates between a local smoothing algorithm and optimization of the master SCOP problem. Each iteration starts with a schedule of planned order releases for the entire supply network. For a subset of PUs, the order release schedule is then evaluated for lead time feasibility locally (i.e. for each PU separately). If necessary, local adjustments are made to the schedule. Next, the locally adjusted schedules are translated to additional constraints in the master SCOP problem which is subsequently reoptimized. The iteration continues for another subset of PUs until each PU has been visited once. The LTA procedure is graphically shown in Fig. 2.

The section is organized as follows: First we formulate the anticipation function mathematically. Then we describe the local smoothing algorithm. Next we discuss in which order PUs are visited and how the locally smoothed schedules are translated into additional constraints in the master SCOP model. Finally, we discuss two strategies for dealing with the situation where no lead time feasible plan is possible.

4.1 Anticipation model

The SCOP function controls the flow of goods in the supply chain network through the release of production orders to the PUs. The planned lead time specifies the amount of time that is allowed to the PU to process orders. The PU may only be expected to observe these planned lead times if the cumulative processing time required to finish these orders does not exceed the planned lead time. For the deterministic anticipation function, this requirement is formalized by constraints (10)–(12). If processing times are stochastic, constraint (12) may overestimate the output rate because it fails to account for the increased congestion in the PU that is caused by stochastic processing times.

We now formulate the stochastic equivalent of Eqs. (10)–(12). To avoid making assumptions on the order of processing, we consider the aggregate workload $V_k(t)$ in

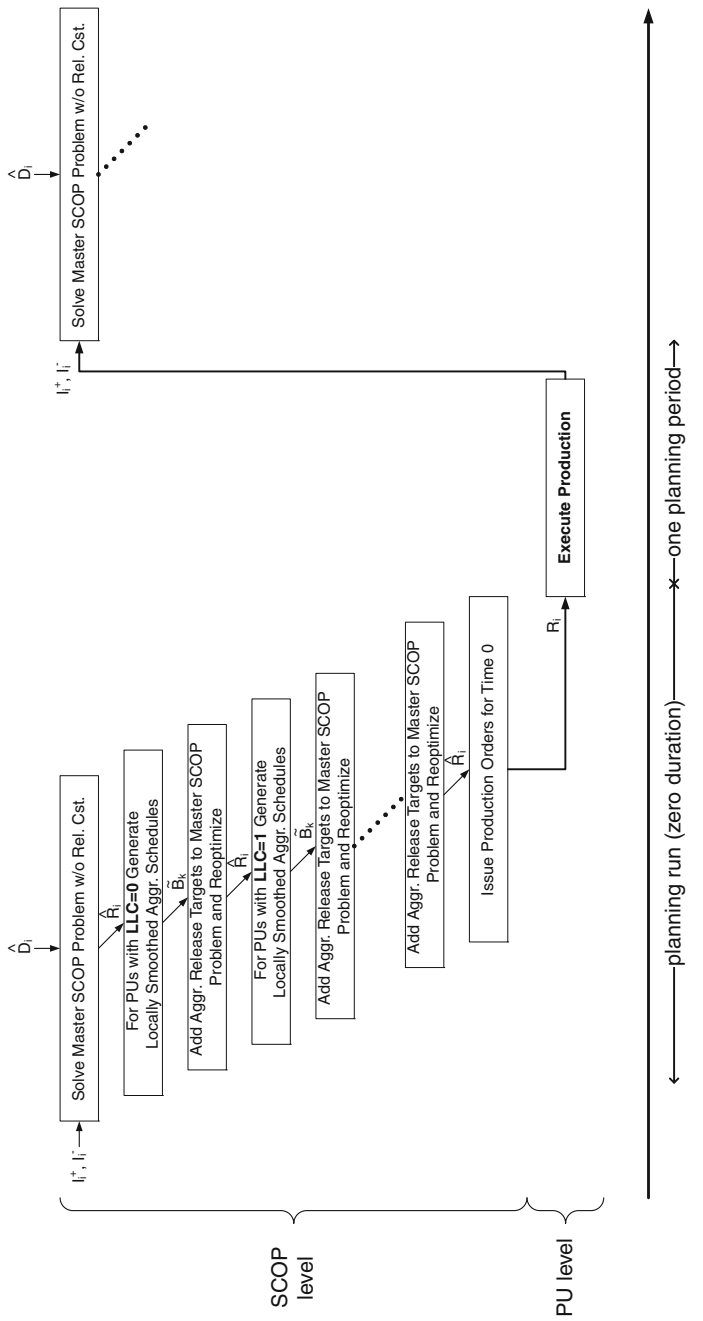

Fig. 2 The lead time anticipation procedure

the PUs as defined in Eq. (2). Similarly, we consider the aggregate released workload $B_k(t)$ as defined in Eq. (3). Without loss of generality we assume that the amount of processing time available in a period is $C = 1$. The dynamics of the workload process are described by the following equation:

$$V_k(t) = (V_k(t-1) - 1)^+ + B_k(t), \qquad k = 1, 2, \ldots, \ t = 0, 1, \ldots, H-1 \tag{17}$$

That is, the workload at the start of a period is equal to any residual workload from the previous period which is the surplus of that period's workload over the capacity available, increased with the newly released workload at the start of the period.

Our interest is to obtain a schedule of planned order releases that is lead time feasible where we define lead time feasibility to be the event that the WIP in a PU can be cleared within the planned lead time. Note that lead time feasibility depends only on the workload at the start of a period and not on the releases after that period since we assume priority is given to jobs released in an earlier period. If the cumulative processing time required to finish all the work in the PU at some time t is less than the available processing time in the planned lead time, then the schedule of planned order releases is lead time feasible for time t. If it is lead time feasible for all times t in the horizon, then the schedule is lead time feasible. Because orders are released to a PU only at the start of a planning period, it suffices to require that

$$V_k(t) \leq L_k$$

However, a constraint on the workload in this form would not be very meaningful since we do not know the realization of $V_k(t)$ a priori. Alternatively, we may specify that the expected workload must satisfy the lead time constraint. That is,

$$\mathbb{E}[V_k(t)] \leq L_k, \qquad k = 1, 2, \ldots, \ t = 0, 1, \ldots, H-1 \tag{18}$$

However, this approach is not very robust since there is a considerable probability that $V_k(t)$ will exceed its expected value. We therefore formulate the constraint as follows:

$$\mathbb{P}\{V_k(t) \leq L_k\} \geq \phi, \qquad k = 1, 2, \ldots, \ t = 0, 1, \ldots, H-1 \tag{19}$$

We have thus introduced a new parameter ϕ to the SCOP model that specifies the desired lead time reliability of the production plan. The higher its value, the smaller the probability that production orders will be tardy in a period. The flip-side of the coin is that the higher reliability comes at the cost of a more restrictive workload constraint which limits the expected output rate. Tardiness and expected output have opposite effects on the total inventory level. This influence is studied in the simulation experiments which are described in Sect. 5.

4.2 Local smoothing algorithm

With constraints (10) to (12) replaced by their stochastic counterparts (17) and (19), at the current state of the art, the SCOP problem becomes mathematically intractable

for any realistically sized instance. Therefore, we present a heuristic procedure for solving the SCOP problem. In this subsection we present the algorithm for smoothing the schedule of planned order releases for a single PU. First, however, we discuss how to track the dynamics of the workload described by Eq. (17).

4.2.1 Tracking of the workload in the PU

Exact evaluation of Eq. (17) is not possible in general. Therefore, we deploy the moment-iteration method (cf. De Kok 1989) to track mean and variance of the workload over time. The moment-iteration method is based on the assumption that the workload released in a period is independent of the residual workload from the previous period such that variances may simply be added. The moment-iteration algorithm is as follows:

Moment-iteration algorithm

1. Start with $\mathbb{E}[V_k(0)] := \mathbb{E}[V_k(0-)] + \mathbb{E}[B_k(0)]$ and $\mathrm{Var}[V_k(0)] := \mathrm{Var}[V_k(0-)] + \mathrm{Var}[B_k(0)]$, $t := 0$
2. Fit a Gamma distribution to the mean and variance of $V_k(t)$
3. Calculate $\mathbb{E}[V_k(t+1)] := \mathbb{E}\left[(V_k(t)-1)^+\right] + \mathbb{E}[B_k(t+1)]$ and $\mathrm{Var}[V_k(t+1)] := \mathrm{Var}\left[(V_k(t)-1)^+\right] + \mathrm{Var}[B_k(t+1)]$
4. If $t < H-1$ set $t := t+1$ and goto 2, else stop.

The mean and variance of $V_k(0-)$ are given by

$$\mathbb{E}[V_k(0-)] = \sum_{i \in U_k} W_i(0-)\,\mu_i, \tag{20}$$

$$\mathrm{Var}[V_k(0-)] = \sum_{i \in U_k} W_i(0-)\,\sigma_i^2, \tag{21}$$

and the mean and variance of $B_k(t)$ are given by

$$\mathbb{E}[B_k(0)] = \sum_{i \in U_k} R_i(0)\,\mu_i, \tag{22}$$

$$\mathbb{E}[B_k(t)] = \sum_{i \in U_k} (\hat{R}_i(t) + \mathbb{E}[\epsilon_i(t)])\,\mu_i, \qquad t = 1, 2, \ldots, H-1, \tag{23}$$

$$\mathrm{Var}[B_k(0)] = \sum_{i \in U_k} R_i(0)\,\sigma_i^2, \tag{24}$$

$$\mathrm{Var}[B_k(t)] = \sum_{i \in U_k} \left(R_i(t)\,\sigma_i^2 + \mu_i^2\,\mathrm{Var}\left[\epsilon_{i,t}\right]\right), \qquad t = 1, 2, \ldots, H-1 \tag{25}$$

Due to the intricate way that it is influenced by deviations of the actual state from the planned state, $\epsilon_i(t)$ is not easily characterized precisely. We imagine that it is measured directly in practice, but here we propose the following approximation:

$$\mathbb{E}[\epsilon_i(t)] = 0, \qquad \mathrm{Var}[\epsilon_i(t)] = \mathrm{Var}[D_i(t-1)] + \sum_j \breve{a}_{ij}^2\,\mathrm{Var}\left[D_j(t-1)\right], \tag{26}$$

where $\breve{a}_{ij}^2 = \sum_{n \in U} a_{in}^2 \breve{a}_{nj}^2$. That is, the planned order releases are unbiased and the deviation is equal to the echelon demand forecast error of the previous period. This approximation is motivated by the idea that deviations of the state in periods before time $t-1$ are accounted for in the release decisions in periods before time t.

Finally, the mean and variance of the residual workload are calculated from the Gamma distribution fitted in step 2. Let α and θ be the parameters of the fitted Gamma distribution such that $\mathbb{E}[V_k(t)] = \alpha\,\theta$ and $\mathrm{Var}[V_k(t)] = \alpha\,\theta^2$. Then, letting $\bar{G}_{\alpha,\theta}$ be the complement distribution function, we have

$$\mathbb{E}\left[(V(t)-1)^+\right] = \alpha\,\theta\,\bar{G}_{\alpha+1,\theta}(1) - \bar{G}_{\alpha,\theta}(1), \tag{27}$$

$$\mathrm{Var}\left[(V(t)-1)^+\right] = (\alpha+1)\,\alpha\,\theta^2\,\bar{G}_{\alpha+2,\theta}(1) - 2\alpha\,\theta\,\bar{G}_{\alpha+1,\theta}(1) + \bar{G}_{\alpha,\theta}(1) - \mathbb{E}\left[(V(t)-1)^+\right]^2 \tag{28}$$

The derivation of these formulas can be found in Appendix A.

4.2.2 Smoothing procedure

If a schedule of planned order releases is not lead time feasible, then planned order releases in the period where the infeasibility occurs need to be reduced and the surplus must be covered by order releases in another period. Since backordering costs are assumed to be (much) higher than inventory holding costs, it is clear that the period to which planned order releases are rescheduled must precede the period where they were originally planned. Preferably, they are rescheduled to an adjacent period such that the amount of time that the items need to be kept in stock upon finishing is minimized. The rescheduling of planned order releases is not trivial since the additional planned order releases not only affect the workload in the period to which they are rescheduled, but potentially also the workload in subsequent periods.

In this subsection, we discuss a heuristic procedure for smoothing a schedule of planned order releases in order to make it lead time feasible. In the smoothing procedure we focus on the aggregate workload dynamics in the PU. We do not specify which items need to be rescheduled at this point. In order to take into account material availability and the cost structure of various items, detailed rescheduling is done in the reoptimization of the SCOP problem that follows the local smoothing procedure. In order to formally define rescheduling, the following assumption is key to this subsection. We assume that if a proportion of the aggregate workload released in a period is rescheduled to an earlier period, then both the coefficient of dispersion (CD) for the workload that is being rescheduled and the CD of the workload that remains scheduled in the original period, equal the CD of the workload in the original schedule. That is, if $B_k(t) = \tilde{B}_k(t) + \breve{B}_k(t)$, where $\tilde{B}_k(t)$ and $\breve{B}_k(t)$ are the workloads remaining and being rescheduled, respectively, then

$$\frac{\mathrm{Var}[B_k(t)]}{\mathbb{E}[B_k(t)]} = \frac{\mathrm{Var}\left[\tilde{B}_k(t)\right]}{\mathbb{E}\left[\tilde{B}_k(t)\right]} = \frac{\mathrm{Var}\left[\breve{B}_k(t)\right]}{\mathbb{E}\left[\breve{B}_k(t)\right]}$$

The CD assumption allows us to describe rescheduling of aggregate workload in mathematical terms. We say that a proportion $p, 0 \leq p \leq 1$ of $B_k(t)$ is rescheduled to period $t-1$ if this results in the adjusted aggregate released workloads $\tilde{B}_k(t-1)$, $\tilde{B}_k(t)$ satisfying

$$\mathbb{E}\left[\tilde{B}_k(t-1)\right] = \mathbb{E}\left[B_k(t-1)\right] + p\,\mathbb{E}\left[B_k(t)\right], \tag{29}$$

$$\mathbb{E}\left[\tilde{B}_k(t)\right] = (1-p)\,\mathbb{E}\left[B_k(t)\right], \tag{30}$$

$$\mathrm{Var}\left[\tilde{B}_k(t-1)\right] = \mathrm{Var}\left[B_k(t-1)\right] + p\,\mathrm{Var}\left[B_k(t)\right], \tag{31}$$

$$\mathrm{Var}\left[\tilde{B}_k(t)\right] = (1-p)\,\mathrm{Var}\left[B_k(t)\right] \tag{32}$$

The following two observations are key to the local smoothing algorithm that we discuss next:

1. **(Monotonicity)** The function $L^{\phi}\left(V_k(t)\right) := \min\{l : \mathbb{P}\{V_k(t) \leq l\} \geq \phi\}$ is non-increasing in the rescheduling proportion p.

   Proof Consider realizations of $V_k(t-1)$ and $B_k(t)$ and select two scalars $\tilde{p}, \check{p}$ such that $0 \leq \tilde{p} \leq \check{p} \leq 1$. Then, $\check{V}_k(t), \tilde{V}_k(t)$ are given by $\check{V}_k(t) := \max\{V_k(t-1) + \check{p}\,B_k(t) - 1, 0\} + (1-\check{p})\,B_k(t)$, and $\tilde{V}_k(t) := \max\{V_k(t-1) + \tilde{p}\,B_k(t) - 1, 0\} + (1-\tilde{p})\,B_k(t)$. The proof follows directly from the fact that $\check{V}_k(t) \leq \tilde{V}_k(t)$:

$$\begin{aligned}
\check{V}_k(t) - \tilde{V}_k(t) &= \max\{V_k(t-1) + \check{p}\,B_k(t) - 1, 0\} + (1-\check{p})B_k(t) \\
&\quad - (\max\{V_k(t-1) + \tilde{p}\,B_k(t) - 1, 0\} + (1-\tilde{p})B_k(t)) \\
&= \max\{V_k(t-1) + \check{p}\,B_k(t) - 1 - \check{p}\,B_k(t), -\check{p}\,B_k(t)\} \\
&\quad - \max\{V_k(t-1) + \tilde{p}\,B_k(t) - 1 - \tilde{p}\,B_k(t), -\tilde{p}\,B_k(t)\} \\
&= \max\{V_k(t-1) - 1, -\check{p}\,B_k(t)\} - \max\{V_k(t-1) - 1, -\tilde{p}\,B_k(t)\}
\end{aligned}$$

   Since $-\check{p}\,B_k(t) \leq -\tilde{p}\,B_k(t)$, we have

$$\max\{V_k(t-1) - 1, -\check{p}\,B_k(t)\} \leq \max\{V_k(t-1) - 1, -\tilde{p}\,B_k(t)\}$$

   and therefore $\check{V}_k(t) - \tilde{V}_k(t) \leq 0 \equiv \check{V}_k(t) \leq \tilde{V}_k(t)$.

2. **(Bounded Residual Work)** Residual workload increases if we reschedule a proportion of work from period t to period $t-1$. However, the increase of the residual workload $\left(\tilde{V}_k(t-1) - 1\right)^+$ is bounded above because, by constraint (19), $\tilde{V}_k(t-1)$ is bounded. Suppose that initially $L^{\phi}\left(V_k(t-1)\right) < L$, $\left(\tilde{V}_k(t-1) - 1\right)^+$ will first increase with p up to the point where constraint (19) is tight. After that point, work from period $t-1$ will be rescheduled to earlier periods such that constraint (19) remains satisfied with equality, and $\left(\tilde{V}_k(t-1) - 1\right)^+$ approximately remains unchanged with a further increase of p.

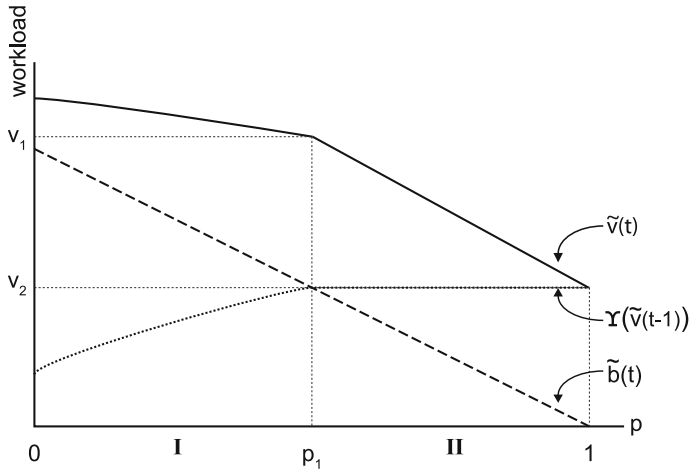

Fig. 3 Workload smoothing heuristic

(There may be a change of mix due to rescheduling resulting in minor variations in the mean and variance of the workload.)

For clarity of the formulation of the local smoothing algorithm it is helpful to use the following vector notation:

$$\mathbf{v}(t) = \begin{pmatrix} \mathbb{E}\left[V_k(t)\right] \\ \mathrm{Var}\left[V_k(t)\right] \end{pmatrix}, \qquad \mathbf{b}(t) = \begin{pmatrix} \mathbb{E}\left[B_k(t)\right] \\ \mathrm{Var}\left[B_k(t)\right] \end{pmatrix}, \qquad \mathbf{v}(0-) = \begin{pmatrix} \mathbb{E}\left[V_k(0-)\right] \\ \mathrm{Var}\left[V_k(0-)\right] \end{pmatrix},$$

where we have dropped the index k since for the local smoothing algorithm, we consider a PU in isolation. We also introduce the functions

$$\Upsilon\left(\mathbf{v}(t)\right) = \begin{pmatrix} \mathbb{E}\left[(V_k(t)-1)^+\right] \\ \mathrm{Var}\left[(V_k(t)-1)^+\right] \end{pmatrix}$$

and

$$L^{\phi}\left(\mathbf{v}(t)\right) = \min\left\{l : \mathbb{P}\left\{\Gamma_{\mathbf{v}(t)} \le l\right\} \ge \phi\right\}$$

where $\Gamma_{\mathbf{v}(t)}$ is a Gamma random variable with expectation and variance specified by $\mathbf{v}(t)$. The latter function gives the minimum lead time within which a workload specified by $\mathbf{v}(t)$ can still be processed. Note that the CD assumption allows us to describe a rescheduling action simply in terms of scalar multiplications, additions, and subtractions. That is, rescheduling a proportion p of period t to period $t-1$ yields the adjusted aggregate release amounts

$$\tilde{\mathbf{b}}(t-1) = \mathbf{b}(t-1) + p\mathbf{b}(t), \qquad \tilde{\mathbf{b}}(t) = (1-p)\mathbf{b}(t).$$

Figure 3 explains the main idea behind the smoothing algorithm. The rescheduling proportion p is set out on the horizontal axis. The three curves represent the

residual workload $\Upsilon(\tilde{\mathbf{v}}(t-1))$, the workload released at the start of the period $\tilde{\mathbf{b}}(t)$, and their sum which gives the total workload at the start of the period $\tilde{\mathbf{v}}(t) = \Upsilon(\tilde{\mathbf{v}}(t-1)) + \tilde{\mathbf{b}}(t)$. We have plotted only the expectation dimension. Note that the curve for $\Upsilon(\tilde{\mathbf{v}}(t-1))$ is not linear in region I since the function Υ is not linear in p. The graph for the variance is similar to the graph for the expectation.

Residual workload is the amount of work that is not completed at the end of period $t-1$ and is carried over to period t. As the amount of workload that is rescheduled from period t to period $t-1$ increases, the residual workload initially also increases. Due to the Bounded Residual Work property, there is a point p_1 where the residual work no longer increases with a further increase of p. Since the amount of workload released does decrease with a further increase of p, we observe a steeper decrease of the total workload after point p_1.

Starting at the end of the planning horizon and working backward through the aggregate release schedule, the smoothing algorithm every time satisfies constraint (19). If the constraint is satisfied for period t it continues with the period $t-1$. Otherwise, p is increased for that period until constraint (19) is satisfied. The search for the optimal p^* for a period t is conducted in two steps. First, a search direction is determined depending on the whether p^* lies in region I or in region II of the graph. Next, a line simple search is used to find the p that and corresponding $\tilde{\mathbf{v}}(t), \tilde{\mathbf{b}}(t)$ such that $L^{\phi}(\tilde{\mathbf{v}}(t)) = L$. (In our implementation of the algorithm Brent's method is used.) The pseudo-code for the local smoothing algorithm is as follows:

Local Smoothing Algorithm

1. Initialize with $\tilde{\mathbf{b}}(t) := \mathbf{b}(t)$, for t=0,1,…,H-1
2. Set $\tilde{\mathbf{v}}(0) := \mathbf{v}(0) = \mathbf{v}(0-) + \mathbf{b}(0)$ and for each period $t = 1, 2, \ldots, H-1$ calculate $\tilde{\mathbf{v}}(t) := \mathbf{v}(t) = \Upsilon(\mathbf{v}(t-1)) + \mathbf{b}(t)$
3. Set $t := H-1$
4. If $L^{\phi}(\tilde{\mathbf{v}}(t)) \le L$, then goto step 9
5. If $L^{\phi}(\tilde{\mathbf{v}}(t-1)) > L$, then
   $\bar{p} := \max\left\{p : L^{\phi}((1+p)\cdot\tilde{\mathbf{v}}(t-1)) < L, -1 \le p < 0\right\}$,
   and $\bar{\mathbf{v}}(t-1) := (1+\bar{p})\cdot\tilde{\mathbf{v}}(t-1)$
   else
   $\bar{p} := \min\left\{p : L^{\phi}(\tilde{\mathbf{v}}(t-1) + p\cdot\mathbf{b}(t)) \ge L, 0 \le p \le 1\right\}$,
   and $\bar{\mathbf{v}}(t-1) := \tilde{\mathbf{v}}(t-1) + \bar{p}\cdot\mathbf{b}(t)$
6. Set $p_1 := \max\{0, \bar{p}\}$, $\mathbf{v_1} := \Upsilon(\tilde{\mathbf{v}}(t-1)) + (1-p_1)\mathbf{b}(t)$.
7. If $L^{\phi}(\mathbf{v_1}) \le L$, then
   $p^* := \min\left\{p : 0 \le p \le p_1,\ L^{\phi}\left(\Upsilon\left[\tilde{\mathbf{v}}(t-1) + p\,\mathbf{b}(t)\right] + (1-p)\mathbf{b}(t)\right) \le L\right\}$
   else
   $p^* := \min\left\{p : p_1 < p \le 1,\ L^{\phi}\left(\Upsilon\left[\tilde{\mathbf{v}}(t-1)\right] + (1-p)\mathbf{b}(t)\right) \le L\right\}$
8. Set $\tilde{\mathbf{b}}(t-1) := \tilde{\mathbf{b}}(t-1) + p^*\cdot\tilde{\mathbf{b}}(t)$, $\tilde{\mathbf{v}}(t-1) := \tilde{\mathbf{v}}(t-1) + p^*\cdot\tilde{\mathbf{b}}(t)$, and $\tilde{\mathbf{b}}(t) := (1-p^*)\cdot\tilde{\mathbf{b}}(t)$
9. If $t > 1$ then $t := t-1$, goto 4 else goto 10
10. Set $\bar{p} := \min\left\{p \ge 0 : L^{\phi}\left((1-p)\cdot\tilde{\mathbf{b}}_0\right) \le L\right\}$, $\tilde{\mathbf{b}}(-1) := \bar{p}\cdot\tilde{\mathbf{b}}(0)$, $\tilde{\mathbf{b}}(0) := (1-\bar{p})\cdot\tilde{\mathbf{b}}(0)$
11. Stop.

The algorithm initializes in steps 1 and 2 where $\mathbf{v}(t)$ and $\Upsilon(\mathbf{v}(t))$ for $t = 0, 1, \ldots, H-1$ are calculated for the original aggregate workload schedule. Starting with the last period in the horizon, it is verified whether the total workload can

be cleared within the lead time in step 4. If the workload cannot be cleared within the planned lead time L, then the point p_1 that separates region I and region II is determined in step 5 and 6. There are two possible scenarios. If the workload in the preceding period $\tilde{\mathbf{v}}(t-1)$ already exceeds L, then $p_1 = 0$. In that case, the helper variable $\bar{p}$ is the fraction of $\tilde{\mathbf{v}}(t-1)$ that can be cleared within the planned lead time and $\bar{\mathbf{v}}(t-1)$ is corresponding workload. If the workload in the preceding period can be cleared in less than the planned lead time, then $p_1 > 0$ is the maximum fraction of $\tilde{\mathbf{b}}(t)$ that can be added to $\tilde{\mathbf{v}}(t-1)$ without violating the lead time constraint and the corresponding workload is $\bar{\mathbf{v}}(t-1)$.

In step 7 of the algorithm, it is tested whether the point p^* lies in region I or II by considering whether the total workload $\mathbf{v}_1$ at point p_1 can still be cleared in the planned lead time. If p^* lies in region I (i.e. $L^{\phi}(\mathbf{v}_1) < L$), then the residual workload depends on the p as specified in the first part of step 7. Otherwise, due to the Bounded Residual Work property, the residual work is $\Upsilon(\tilde{\mathbf{v}}(t-1))$ independent of the exact rescheduling proportion (part 2 of step 7). In step 8, adjusted variables are updated based on the rescheduling proportion. The iteration then continues with the preceding period until all periods have been visited. A special case is period $t = 0$ where there is no preceding period. The overflow workload is defined as the amount which the workload in period $t = 0$ must be reduced in order for the schedule to be lead time feasible. This amount is determined in step 10 and is stored in the variable $\tilde{\mathbf{b}}(-1)$.

The result of the smoothing algorithm is an aggregate release schedule that satisfies the lead time feasibility constraint (19) for PU k in all periods. A lead time feasible schedule may not be possible at all time in which case there is an overflow amount of workload which is given by $\tilde{\mathbf{b}}(-1)$. In Sect. 4.3.1 we will discuss two strategies for dealing with the overflow workload.

4.3 Updating the master SCOP problem

The local smoothing algorithm produces aggregate release schedules for individual PUs that are lead time feasible. In the previous section, we have omitted the index k for the vectors $\tilde{\mathbf{b}}(t)$ because there a local smoothing procedure is discussed for a single PU. Here we introduce it again so the smoothed aggregate release schedule for PU k is given by the sequence $(\tilde{\mathbf{b}}_k(0), \ldots, \tilde{\mathbf{b}}_k(H-1))$. The expectation part of the adjusted aggregate release amounts $\tilde{\mathbf{b}}_k(t)$ is added as a target to the master SCOP problem. These targets affect the timing of order releases and therefore also the timing of dependent requirements and availability of materials at other PUs. Since the smoothing works in one direction only (i.e. aggregate release amounts may be rescheduled to earlier periods but never to later periods), only the effect of the targets on the upstream PUs needs to be taken into account. Note that earlier availability of materials will not result in earlier production in downstream PUs since this would merely imply a repositioning of "smoothing inventory" to downstream stock points where the holding costs are higher.

The feedforward property of the supply network allows us to conduct the local smoothing and reoptimization of the master SCOP problem in steps (see Fig. 2), each time taking into account the changes in dependent requirements. In the LTA procedure PUs are visited in increasing order of their PU low-level-code (LLC). The PU LLC is

the maximum of the LLCs of items produced in the PU:

$$\text{LLC} = \max_{i \in U_k} \{v_i\} \tag{33}$$

where v_i is defined as

$$v_i := \begin{cases} 0, & \text{if } U_S(i) = \emptyset \\ 1 + \max_{j \in U_S(i)} \{v_j\}, & \text{otherwise} \end{cases} \tag{34}$$

The LTA procedure is illustrated in Fig. 2. Initially, the master SCOP problem does not contain any smoothing targets and consists of Eqs. (7)–(9) only. This LP problem is solved to obtain an initial schedule of planned order releases. Next, the smoothing algorithm is applied to produce aggregate release schedules for PUs with external demand only (i.e. PUs with LLC = 0). These aggregate release schedules are used to generate the first set of release targets for the master SCOP problem which is subsequently reoptimized to obtain a new schedule of planned order releases for the entire supply network. Next, the smoothing algorithm is applied for those PUs that only have dependent demand from the PUs with LLC = 0 (and possibly some external demand as well). The LTA procedure continues in this manner until all PUs have been visited once and a complete set of aggregate release targets is generated and the master SCOP problem is optimized for the supply network.

We now formulate the aggregate release targets for the master SCOP problem. The aggregate release targets are formulated in way such that the following two requirements are met:

- The local smoothing algorithm does not take into account the availability of materials and other resources in the supply network. For this reason, it may not be possible during reoptimization of the master SCOP problem to obtain a schedule of planned order releases that conforms to aggregate release targets. Rather than formulating hard constraints, violation of the targets is penalized to prevent the master SCOP problem from becoming infeasible.
- The aggregate release target is met by rescheduling the original planned order releases. The targets may not be obtained by planning order releases for items or quantities that are not required.

Let $\hat{B}_k(t)$ be the expected aggregate release amount corresponding to the initial schedule of planned order releases, calculated via Eqs. (22) and (23), and let $\hat{V}_k(0-)$ be the initial expected workload calculated via Eq. (20). Furthermore, let $\tilde{B}_k(t) := (\tilde{\mathbf{b}}_k(t))_1$ be the expected workload in the adjusted aggregate release schedule produced by the local smoothing algorithm and let $\tilde{R}_i(t)$ denote the adjusted order release quantity.

Figure 4 shows an example of an initial aggregate release schedule (solid line) and an adjusted aggregate release schedule (dotted line). We denote the (vertical) difference between the two lines for period t by $\delta_k(t) := \tilde{B}_k(t) - \hat{B}_k(t)$ and the cumulative difference by $\Delta_k(t) := \sum_{s=0}^{t} \delta_k(t)$. In the light gray area, adjusted releases exceed the original releases ($\delta_k(t) > 0$), and in the dark gray area the original releases exceed

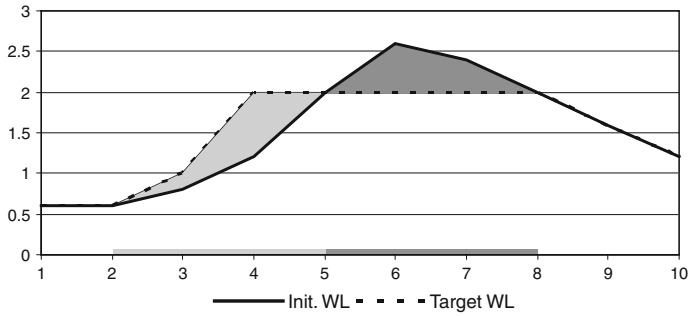

Fig. 4 Residual and target workload

the adjusted releases ($\delta_k(t) < 0$). After the dark gray area, the two schedules coincide again ($\Delta_k(t) = 0$).

The following constraints are added to the master SCOP problem after each local smoothing step:

$$\sum_{s=0}^{t} \mu_i \left(\tilde{R}_i(s) - \hat{R}_i(s)\right) + \mu_i \, O_i(t) = Z_i(t) \, \Delta_k(t), \tag{35}$$

$$\sum_{i \in U_k} Z_i(t) = 1 \tag{36}$$

These constraints are formulated such that the adjusted release decisions follow the aggregate release schedule as closely as possible. $Z_i(t)$ is the allocation of the cumulative difference in Fig. 4 to individual products and $O_i(t)$ is the quantity of product i that is produced less corresponding to this allocation. That is, we do not make this allocation a priori but leave it as a decision variable for the goods flow model. There are a number of useful properties to the above formulation of the constraints:

- There is a time u where $\Delta_k(u) = 0$. At this point, $\sum_{s=0}^{u} \tilde{R}_i(s) \leq \sum_{s=0}^{u} \hat{R}_i(s)$. That is, cumulative releases in the adjusted plan cannot exceed the cumulative releases in the original plan. So there can be no decision to release orders for items that are not required according to the original plan.
- $O_i(t) \geq 0$ is an auxiliary variable that relaxes constraint (35). This variable prevents the goods flow model from becoming infeasible if materials required in production cannot be made available earlier, whereby forward rescheduling of release quantities becomes impossible. We penalize the use of this slack variable by adding to the objective the term $\sum_{t=0}^{H-1} \sum_k \sum_{i \in U_k} c_i^b \, O_i(t)$. The choice of the penalty variable c_i^b is somewhat arbitrary. It should be set (much) larger than the holding cost.

Remark 1 (A note on the prioritization of order releases) A useful by-product of the LTA procedure is a prioritization of order releases. The initial schedule of planned order releases contains the latest production starts that, given the planned lead time, meet the requirements. Planned order releases that are rescheduled to earlier periods

as a consequence of the aggregate release targets wait in the downstream stock point for their requirement upon completion. Compared with the order releases that were initially planned in these periods, the rescheduled order releases thus have a lower priority. The PU should therefore give priority to the order releases $\tilde{R}_i^p(t)$ defined recursively by

$$\tilde{R}_i^p(t) := \min\left\{\sum_{s=0}^{t} \hat{R}_i(s), \sum_{s=0}^{t} \tilde{R}_i(s)\right\} - \sum_{s=0}^{t-1} \tilde{R}_i^p(s), \quad \text{for all } i \in U,\ t = 0, 1, \ldots, H-1 \tag{37}$$

That is, for some period $t \geq 0$ the cumulative planned order releases in the initial schedule up to period t have priority and need to be completed first. Any additional rescheduled releases are not immediately required after their planned lead time and have a lower priority.

4.3.1 Overloading

The existence of overflow workload (i.e. $\tilde{\mathbf{b}}_k(-1) > \mathbf{0}$) indicates that no planned lead time feasible workload schedule can be obtained by rescheduling releases to earlier periods only. Here we discuss two strategies for dealing with a positive overload. In the *conservative strategy*, we discard the overflow workload such that the resultant schedule of planned order releases is lead time feasible. This strategy implies that the total order release quantity in the final schedule may be less than in the initial schedule. Through the inventory balance equations in the master SCOP problem, these changes propagate to the supply of the end-items. That is, the resultant schedule is lead time feasible but plans for higher backorders of end-items. These (temporary) deficits in supply over demand must be covered by the safety stocks for end-items.

In the *overloading strategy* we temporarily allow lead time constraint (19) to be violated and maintain workload levels in the PU that cannot reliably be cleared within the planned lead time. Hence the term "overloading". From the literature on clearing functions we learn that a higher workload results in a higher expected output rate, and thus a faster reduction of any backorders. On the other hand, overloading leads to tardiness of production orders. Not only does tardiness result in delayed supply of end-items, but it may also introduce planning inefficiencies. Particularly in assembly networks, tardiness may lead to remnant stocks of other materials in downstream stages.[1] Tardiness may also lead to an unplanned shift of capacity consumption to later periods in downstream stages, leading to lost capacity in the periods where the consumption was originally planned.

We compare the conservative strategy to the overloading strategy in our simulation experiments (see Sect. 5.5). The conservative option appears to restrict the release of orders too much. The overloading strategy gives better results in almost all experi-

[1] Remnant stocks are stocks of other materials required in the downstream assembly step that wait for completion of the tardy order (cf. De Kok 2003).

ments. In the remainder of this section we discuss the overloading option in more detail.

In busy situations, the overflow $\tilde{\mathbf{b}}_k(-1)$ may be high. In these situations, it makes no sense to overload the PU directly with a large amount of work. Such a strategy would lead to large and costly WIP levels without significant reduction of the backlog. We can obtain an upper bound $W^*(t)$ on the cost-optimal workload in a PU that faces a backorder situation in the following way: Consider a single item setting where the number of backorders is large. That is, $I^-(t) > 0$ for all $W(t)$.

$$\begin{aligned} W^*(t) &= \arg\min c^w\, W(t) + c^b \mathbb{E}\left[I^-(t) | W(t),\, I^-(t) > 0\right] \\ &= \arg\min c^w\, W(t) + c^b \mathbb{E}\left[(I(t-1) - D(t-1) + Y(t))^- | W(t),\, I^-(t) > 0\right] \\ &= \arg\min c^b \mathbb{E}\left[D(t-1) - I(t-1)\right] + c^w\, W(t) - c^b \mathbb{E}\left[Y(t) | W(t)\right] \end{aligned} \tag{38}$$

The first term on the right-hand side of Eq. (38) does not depend on $W(t)$ and can be omitted. Using furthermore the proportionality between WIP and FGI holding costs, and backordering costs, and defining

$$f^{wb} = \frac{c^w}{c^b} = \frac{f^{wh}}{f^{bh}}$$

we have

$$W^*(t) = \arg\min f^{wb}\, W(t) - \mathbb{E}\left[Y(t) | W(t)\right], \tag{39}$$

$W^*(t)$ is an upper bound on the economical amount of WIP in the PU. Beyond this quantity, the marginal cost of holding WIP exceeds the marginal expected reduction of backordering costs. In our multi-item setting the cost trade-off is less obvious since the WIP may be composed of different item types. As an approximation, we propose to substitute workload for WIP and replace output by the amount of workload that is cleared in the planned lead time. The workload is expressed as the sum of the workload in the lead time feasible plan $\tilde{V}_k(0)$ and a proportion p of the overflow $\tilde{B}_k(-1)$. The optimal overloading proportion p_0^* is then given by

$$p_0^* = \arg\min_{0 \le p \le 1} \left[f_k\, \mathbb{E}\left[\tilde{V}_k(0) + p * \tilde{B}_k(-1)\right] - \mathbb{E}\left[\min\{L_k,\, \tilde{V}_k(0) + p * \tilde{B}_k(-1)\}\right]\right], \tag{40}$$

Since the minimum function in the second term is concave, this cost function is convex. Let the total workload for a period be $V_k(t) \sim \text{Gamma}(\alpha, \theta)$ and let $\bar{G}_{\alpha,\theta}$ be the complement Gamma distribution function. The second term of the cost function (40) is

$$\mathbb{E}\left[\min\{\tilde{V}_k(0) + p * \tilde{B}_k(-1), L_k\}\right] = \alpha\, \theta\, G_{\alpha+1,\theta}(L_k) + L_k\, \bar{G}_{\alpha,\theta}(L_k) \tag{41}$$

where

$$\alpha = \frac{\left(\mathbb{E}\left[\tilde{V}_k(0)\right] + p\,\mathbb{E}\left[\tilde{B}_k(-1)\right]\right)^2}{\mathrm{Var}\left[\tilde{V}_k(0)\right] + p\,\mathrm{Var}\left[\tilde{B}_k(-1)\right]},$$

$$\theta = \frac{\mathrm{Var}\left[\tilde{V}_k(0)\right] + p\,\mathrm{Var}\left[\tilde{B}_k(-1)\right]}{\mathbb{E}\left[\tilde{V}_k(0)\right] + p\,\mathbb{E}\left[\tilde{B}_k(-1)\right]}$$

For the derivation of this equation, see Appendix A.

The optimal p_0^* for period 0 is found though a convex optimization routine. If p_0^* is less than 1, then the optimal overloading fraction to period 1, p_1^*, is found in the same way and a part of the overflow workload $p_1^*\,(1 - p_0^*)\tilde{\mathbf{b}}_k(-1)$ is scheduled in period 1, etc.

5 Comparing anticipation models

In this section, we compare production planning algorithms with different anticipation functions using discrete event computer simulation. The performance of an anticipation function is read from its ability to reduce the inventory holding costs subject to meeting the service level constraint. Safety stocks of end-items are deployed to buffer for any remaining supply deficits. Each anticipation function requires different safety stocks. However, safety stock alone is not the appropriate performance measure. One can think of anticipation functions that meet the service constraint without any safety stock being installed (for example one that always plans order releases well ahead of the planned requirement). Such an anticipation function necessarily builds in other forms of slack that may cancel out or exceed the savings in safety stocks. A good anticipation function meets the service level constraint with a minimum investment in total inventory which besides safety stocks includes WIP, finished goods inventory, and all intermediate stocks. The question that we aim to answer via our simulation experiments is

> Which are the differences among the planning algorithms studied, in the total holding cost of inventory required in the supply network to satisfy the non-stockout probability and how do these differences depend on the characteristics of the supply network.

The planning algorithms that we compare are listed in Table 2. The UNC algorithm corresponds most closely to the MRP logic, with the exception that backordering of components is not allowed. We have argued before that this restriction is necessary in order to obtain a complete planning algorithm. The CAP (capacitated) algorithm is the original model as formulated in De Kok and Fransoo (2003) and the CLF algorithm is the planning algorithm with a clearing function anticipation function as described in Sect. 5.1. Finally, two LTA algorithms with conservative and overloading strategies are compared.

Table 2 Anticipation models

Abbreviation	Description
UNC	The uncapacitated SCOP model given by the objective (7) and constraints (8) and (9) (i.e. no anticipation other than a fixed planned lead time)
CAP	The SCOP model with deterministic capacity given by the objective (7) and constraints (8) to (12)
CLF	The SCOP model with clearing functions given by the objective (7), constraints (8) and (9), and the constraints (45) to (46) defined in Sect. 5.1
LTA-OL	The lead time anticipation procedure with overloading as described in Sect. 4.1
LTA-CS	The lead time anticipation procedure without overloading (conservative strategy) as described in Sect. 4.1

The question remains how to set the safety stocks in combination with each anticipation function such that they meet the non-stockout probability service level. We presume that in practice safety stocks are set using rules of thumb or using more sophisticated planning tools and are subsequently fine-tuned by human planners on the basis of actual service level measurements. Clearly such a procedure cannot be replicated in the experimental setting of this paper. Instead, we deploy a very useful procedure for setting safety stocks in discrete event simulations. This procedure is the safety stock adjustment procedure (SSAP) which we discuss in Sect. 5.2. First, however, we discuss the clearing function model that we use for comparison in Sect. 5.1. In Sect. 5.3 we explain the experimental setup for our comparison study and in Sect. 5.5 discuss the results.

5.1 Clearing function approach

The clearing function that we use in this comparison is proposed in chapters 4 and 5 of Selçuk (2007). However, since this is a single-item clearing function, we reformulate it in terms of workload:

$$\begin{aligned}\sum_{i\in U_k}\mu_i\,\hat{X}_i(t)=\hat{Y}_k(t)&=f\left(\sum_{i\in U_k}\mu_i\left(\hat{W}_i(t)+\hat{R}_i(t)\right)\right)\\&=f\left(\hat{V}_k(t)\right)\\&=\mathbb{E}\left[\min\{V_k(t),1\}\right]\end{aligned}\tag{42}$$

This clearing function is approximated by the following piece-wise linear function (cf. Selçuk 2007):

$$f\left(\hat{V}_k(t)\right) = \begin{cases} \hat{V}_k(t), & \text{if } \hat{V}_k(t) \leq w_{k,\varepsilon} \\ \beta_k + \gamma_k\, \hat{V}_k(t), & \text{if } w_{k,\varepsilon} \leq \hat{V}_k(t) \leq w_{k,1} \\ 1, & \text{if } \hat{V}_k(t) \geq w_{k,1} \end{cases}, \tag{43}$$

where

$$\beta_k := w_{k,\varepsilon}\left(1 - \frac{1 - w_{k,\varepsilon}}{w_{k,1} - w_{k,\varepsilon}}\right), \qquad \gamma_k := \frac{1 - w_{k,\varepsilon}}{w_{k,1} - w_{k,\varepsilon}}$$

The parameter ε is described in Selçuk (2007) as "a service level measure for the clearing of the available WIP in the production unit". The point $w_{k,\varepsilon}$ is set in corresponding to this parameter such that $\mathbb{P}\left\{V_k(t) \geq 1 | \hat{V}_k(t) = w_{k,\varepsilon}\right\} = 1 - \varepsilon$. Similar to Selçuk (2007) we conduct our simulation experiments for $\varepsilon = 0.05$ and $\varepsilon = 0.2$. The point $w_{k,1}$ is set such that $f(w_{k,1}) \approx 1$.

We approximate $V_k(t)$ by a Gamma distribution with parameters α_k and θ_k. The clearing function takes as its argument the expected workload. The Gamma distribution is furthermore determined by its CD which corresponds to the parameter θ_k. We set θ_k to be the long-run weighted average of the CD of the individual processing times:

$$\theta_k = \frac{1}{\sum_{j \in U_k} \lambda_j} \sum_{i \in U_k} \lambda_i \frac{\sigma_i^2}{\mu_i}, \tag{44}$$

where λ_i is the long-run average order release quantity for item i.

We use the approach proposed by Asmundsson et al. (2009) to disaggregate the active processing time in a period. They introduce an allocation variable $Z_i(t)$ specifying the proportion of the aggregate output that is of item i and show that the linearization of the clearing function can be formulated as

$$\mu_i\, \hat{X}_i(t) \leq \beta_k\, Z_i(t) + \alpha_k\, \mu_i \left(\hat{W}_i(t) + \hat{R}_i(t)\right), \qquad \text{for all } t, k, i \in U_k \tag{45}$$

$$\mu_i\, \hat{X}_i(t) \leq Z_i(t), \qquad \text{for all } t, i \tag{46}$$

$$\sum_{i \in U_k} Z_i(t) = 1, \qquad \text{for all } t, k \tag{47}$$

Constraints (45)–(47) replace constraint (12) and together with constraints (10) and (11) form the clearing function anticipation function.

5.2 The safety stock adjustment procedure

The SSAP of Kohler-Gudum and De Kok (2002) is a method for setting safety stocks in simulation experiments. The procedure finds the amount of safety stock required to

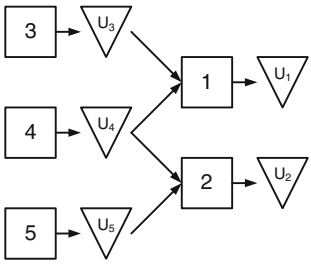

Fig. 5 W-shape supply network

meet a certain service level. The SSAP allows therefore for comparison of planning concepts that keep inventory at different places in the supply network.

The main conceptual idea behind the SSAP is the relative invariance of order release decisions to the safety stock level. If a simulation experiment is replicated with exactly the same parameter setting and random number streams but with the initial stock level and safety stock level for an item changed with the same amount, then the planning algorithm will yield precisely the order release decisions of the first experiment.

The SSAP used in this paper is an adaptation of that described in Kohler-Gudum and De Kok (2002). In the original paper, the service level measure was the fill-rate whereas in this paper we have chosen the non-stockout probability as the service level measure.

The SSAP proceeds as follows: Each simulation experiment is run twice. In the first run, a record is maintained of the lowest net stock level for each end-item in each period. The first run thus provides an empirical distribution of the lowest net stock level. This distribution is shifted by changing the safety stock such that the desired non-stockout probability is obtained. The second run is initiated with initial stock level adjusted by the same amount as the safety stock levels. Then, all decisions from the first run are replayed. The second run is used to obtain the statistics that are of interest.

5.3 Test bed

The test bed for our simulation experiments has a W-shaped topology (see Fig. 5). The W-shape is the simplest generic topology (i.e. combining divergent and convergent goods flows for all downstream stages). The lower echelon contains two stages ($k = 1, 2$) and the upper echelon contains three stages. One upstream stage ($k = 4$) supplies a set U_4 of common components to both downstream stages, whereas the other two ($k = 3, 5$) supply sets U_3 and U_5 of specific components to a single downstream stage only. The behavior of the individual stages in the supply chain is simulated as described in Sect. 2.2.

Even for the simple supply chain topology in Fig. 5, the number of parameters is large. These parameters include the length of the planning horizon, service level, planned lead times, utilization levels, demand and processing time distributions, forecast accuracy, seasonality characteristics, and costs structure. Furthermore, there are parameters specific to the planning model such as the level ϕ for the LTA approach and the level ε for the clearing function approach. Rather than investigating each possible

combination of parameter settings, we randomly select settings from realistic ranges for each experiment. Most parameter settings are drawn from uniform distributions. We shall now discuss parameters of the supply network in detail.

BOM The bill-of-materials is generated in the following way: First the number of end items (ne_k, $k = 1, 2$), and the number of common parts (nc) are drawn. For each end item, a number of specific part types (ns_i) for each end item i is drawn. Let U_E, U_C, and $U_S = \bigcup_{i \in U_E} U_{S_i}$ denote the sets of end items, common items, and specific items, respectively. We then specify the BOM as follows:

$$a_{ij} := \begin{cases} 1, & \text{for all } j \in U_1 \bigcup U_2, i \in U_3 \bigcup U_5 \\ 1, & \text{for all } j \in U_1 \bigcup U_2, i \in U_3 \\ 0, & \text{otherwise} \end{cases}$$

Planned lead time For most experiments we set $L_k = 2$. However, the planned lead time is an important parameter to the SCOP function. As was discussed in Sect. 2.4, it acts as a smoothing mechanism in itself. It is therefore expected that a long planned lead time mitigates the effect of anticipation functions. On the other hand, a short planned lead time may be very restrictive with respect to the workload that is allowed in the PU which may lead to a reduced output. To study these effects, we conduct a series of experiments with a shorter and a longer planned lead time.

Costs The FGI holding cost of items in the upstream stock points is set to one. In each subsequent stage 50% is added to the holding costs. Half of the cost is added in the WIP and the other half is added once the items proceed to the FGI (i.e. $f^{wh} = \frac{5}{6}$). That is,

$$\begin{aligned} c_i^h &= 1, && i \in U_1 \bigcup U_2 \bigcup U_3, \\ c_j^h &= 1.5 \sum_{i \in U} a_{ij}\, c_i^h, && j \in U_1 \bigcup U_2, \\ c_i^w &= \tfrac{5}{6}\, c_i^h, && i \in U \end{aligned}$$

Capacity The utilization rate of resources is likely to have an influence on the benefit of production smoothing. Clearly, if there is never a shortage of capacity, then there is no requirement to smooth the schedule of order releases at all. On the other hand, a high utilization rate may require a constant high level of workload in the PU in order to have sufficient output. A smoothing approach where the workload is restricted may have less favorable results in these situations. In our experiments we study both cases, high and low utilization, separately.

Processing times Processing times μ_i are then drawn from a continuous uniform distribution on $\left[\frac{\rho_k}{|U_k|}(0.5\sqrt{3}), \frac{\rho_k}{|U_k|}(1.5\sqrt{3})\right]$. Subsequently, the processing times are normalized such that $\sum_{i \in U_k} \lambda_i\, \mu_i = \rho_k$. A service time distribution (Ψ_i) is randomly selected for each item among the exponential ($E1$), the Erlang-2 ($E2$), and the

Erlang-4 ($E4$) distribution (corresponding, respectively, in squared coefficients of variation 1, $\frac{1}{2}$, $\frac{1}{4}$).

Demand and forecast Demand is seasonal with a cycle that is the same for all items and has a length that equals the planning horizon. The expected demand for item $i \in U_k,\ k = 1, 2$ in simulation period t (not the period in the planning horizon) is

$$\hat{D}_i(t) = \lambda_i \left(1 + \text{seas} \sin\left(\frac{t}{H}\right) 2\pi\right) \tag{48}$$

where seas defines the amount of seasonality in the demand. The expected demand also is the forecast for the period. We sample the actual demand $D_i(t)$ from a Gamma distribution with expectation $\hat{D}_i(t)$ and a variance of $0.25\,\lambda_i^2$. The actual demand is rounded to the nearest integer, but the integrality difference is added to the next period demand such that the long-term expected demand is consistent with the forecasted demand.

5.4 Simulation environment

The simulation experiments are conducted along the lines of Law (2007) and are implemented in the C# programming language. Since we make pairwise comparisons of planning algorithms, we choose to randomly generate a number of experiments rather than to run a number of replications for a single experiment. An experiment has a warm-up time of 500 periods and additionally a run time of 2000 periods during which statistics are collected. An experiment is repeated for each planning algorithm with common pseudo random number (PRN) streams. A PRN stream is generated for each simulation module (stage in the network) and each period.

The LP problems are solved using the Barrier search implementation of ILOG CPLEX 11.0. Release quantities are translated into a job for each unit. These jobs are then released to the PU in random order. For the LTA approach we do utilize the prioritization discussed in Sect. 4.3: high-priority jobs are released before regular jobs.

5.5 Experiments and discussion

The parameter set for our experiments is specified in Tables 3 and 4. An experiment is repeated for each planning algorithm. There are 40 experimental settings in the first set so there are 400 runs in the first set. The results for set 1 are listed in Table 5. The number in square brackets behind the algorithm name is the setting for the algorithm-specific parameter (ε for CLF and ϕ for LTA). The statistic reported on is the relative difference in total cost between the SCOP approaches listed in Table 2. We set the total cost of CAP to 100. For example, let TC_n^{alg} denote the total cost for an algorithm in the nth experiment; then the average total cost reported for LTA-OL[0.9] is

Table 3 Parameter settings for experiment set 1

Parameter	Description	Range
H	Planning horizon	20
φ	Non-stockout probability	0.98
L_k	Planned lead time	2 periods
$\|U_1\|, \|U_2\|$	Number of end items for stages 1 and 2	1 to 5
$\|U_4\|, \|U_3\|, \|U_5\|$	Number of common items ($k = 4$) and number of specific items ($k = 3, 5$) per for a stage	1 or 2
ρ_k	Utilization rate for stage k	0.75 to 0.85
Ψ_i	Distribution of processing time for item i	$\{E1, E2, E4\}$
$\lambda_i,\ i \in U_E$	Average demand rate for item i	1 to 5
$\mathrm{Var}\left[\epsilon_{i,t}\right]$	Variance of demand and requirements variance estimate	$0.25\,\lambda_i^2$
seas	Seasonality factor	0.4 to 0.8
SCOP algorithms:	UNC, CAP, CLF ($\varepsilon = 0.05, 0.2$), LTA-OL ($\phi = 0.7, 0.8, 0.9$), LTA-CS ($\phi = 0.7, 0.8, 0.9$)	
Number of experiments:	40	

Table 4 Parameter settings for experiment set 2–6

Parameter	Set 2 Short L_k	Set 3 Long L_k	Set 4* High Util.	Set 5 Low Util.	Set 6 No Dem. Unc.
ρ_k	As in set 1	As in set 1	0.85	0.75	As in set 1
$\mathrm{Var}\left[\epsilon_{i,t}\right]$	As in set 1	As in set 1	As in set 1	As in set 1	0
Planned lead time:	1 periods	3 periods	2 periods		
all others as in set 1					
SCOP algorithms:	UNC, CAP, CLF ($\varepsilon = 0.05, 0.2$), LTA-OL ($\phi = 0.7, 0.8, 0.9$), LTA-CS ($\phi = 0.7, 0.8, 0.9$)				
Number of experiments:	20 per set				

* Warm-up of 1000 periods and a run length of 4000 periods for this set

$$\text{Average total cost LTA-OL[0.9]} = \frac{1}{40}\sum_{n=1}^{40}\frac{TC_n^{\text{LTA-OL[0.9]}}}{TC_n^{\text{CAP}}} \times 100$$

The first three columns show the average cost of WIP, finished goods inventory (of $i \in U_k, k = 1, 2$), and the average total cost for the experiments. The columns "Lowest" and "Highest" refer to the lowest and highest total cost observed in all experiments. The column "Best Cases" shows the percentage of experiments where the algorithm had the best performance. The last column but one gives the percentage of the costs that are caused by the safety stock. We note here that this percentage may be more than 100% if the planning algorithm fails to meet the target stock level often

Table 5 Simulation results for set 1

Algorithm	WIP	FGI	Average	Lowest	Highest	Best cases (%)	Sfty as % of total (%)	Fill rate w/o Sfty (%)
CAP	42.51	57.49	100.00	100.00	100.00	0	48.54	48.54
UNC	46.64	61.56	108.20	87.27	121.99	0	30.84	30.84
CLF[0.05]	42.33	93.69	136.02	96.54	1138.14	0	50.40	50.40
CLF[0.2]	39.27	59.16	98.43	82.54	361.08	3	51.22	51.22
LTA-OL[0.7]	43.10	54.89	98.00	79.70	103.94	8	59.73	59.73
LTA-OL[0.8]	43.17	52.89	96.06	77.61	101.36	33	64.35	64.35
LTA-OL[0.9]	43.01	53.13	96.14	72.03	109.70	43	70.29	70.29
LTA-CS[0.7]	42.26	60.44	102.70	93.53	166.49	5	56.06	56.06
LTA-CS[0.8]	42.03	67.26	109.29	94.13	370.28	10	58.47	58.47
LTA-CS[0.9]	41.76	106.16	147.91	94.11	765.87	0	56.04	56.04

and for prolonged periods. In such cases the safety stock required to meet the required service level may be (much) higher than the average physical inventory and may even exceed the average total inventory if other inventories are relatively small.

Both tardiness of orders and uncertainty in the demand are accommodated for by the installment of safety stocks. Safety stock levels are determined through the SSAP such that the non-stockout probability is fixed at 98%. In the last column of Table 5 it is shown which percentage of the customer orders would be satisfied in absence of the safety stock. This percentage expresses the fraction of times that the net stock level in the downstream stock points is no less than the safety stock directly after satisfying a customer order.

The first remarkable observation is that the CLF is not robust. We see that the maximum relative cost for CLF[0.2] is 3.6 times the cost for CAP. For CLF[0.05] the results are even worse. A closer look at the results from the experiments reveals the reason for this poor performance. We see that in some experiments the CLF leads to very high safety stocks. These safety stocks are put in place by the SSAP to ensure that the service level is met. In these cases, the supply is structurally lower than the demand. This is caused by the restriction that the CLF in combination with a planned lead time puts on the workload in the PU: workload is restricted by the amount that the clearing function predicts can be processed within the planned lead time. In turn, the restriction of the workload limits the maximum throughput rate.

We also see the negative effects of the limitation of the workload in the LTA-CS algorithms. Recall that the conservative strategy may limit the total workload in the PU in order to obtain a lead time feasible plan. As the required reliability level ϕ increases for the LTA-CS algorithms, the workload is restricted more and the negative effect on the output becomes sincere.

Interestingly, the effect of the reliability level has an opposite effect for the LTA-OL algorithm with an overloading strategy. The best performance is achieved at levels $\phi = 0.8$ or $\phi = 0.9$. Among these two settings, $\phi = 0.8$ scores best in terms of consistency (its highest and lowest cost are closer). The optimistic LTA-OL algorithm performs better than the conservative LTA-CS algorithm. From this we can deduct that,

Table 6 Simulation results for set 2–3

Algorithm	WIP	FGI	Average	Lowest	Highest	Best cases (%)	Sfty as % of total (%)	Fill rate w/o Sfty (%)
Set 2: short planned lead times								
CAP	16.96	83.04	100.00	100.00	100.00	0	145.89	36.89
UNC	24.62	50.11	74.73	11.70	114.83	3	90.07	14.32
CLF[0.05]	13.91	1445.45	1459.36	480.53	2746.97	0	253.57	28.02
CLF[0.2]	20.24	754.23	774.47	110.52	2470.68	0	228.58	33.41
LTA-OL[0.7]	17.10	47.22	64.32	16.06	95.51	43	65.13	64.72
LTA-OL[0.8]	17.63	44.45	62.08	13.14	97.78	50	37.01	76.35
LTA-OL[0.9]	18.58	55.06	73.63	12.55	122.60	5	24.83	80.89
LTA-CS[0.7]	15.98	338.17	354.14	105.44	696.04	0	232.92	37.13
LTA-CS[0.8]	14.96	850.23	865.20	181.24	4115.50	0	251.17	31.54
LTA-CS[0.9]	13.25	1870.49	1883.74	353.80	6485.89	0	254.95	25.85
Set 3: long planned lead times								
CAP	53.89	46.11	100.00	100.00	100.00	35	49.59	51.34
UNC	55.12	49.56	104.68	100.46	127.27	0	53.01	39.46
CLF[0.05]	53.89	46.16	100.05	97.29	107.16	23	49.37	52.14
CLF[0.2]	51.58	44.23	95.80	80.90	101.83	13	49.52	51.85
LTA-OL[0.7]	54.31	46.92	101.24	92.20	105.04	3	46.21	56.35
LTA-OL[0.8]	54.40	46.53	100.93	91.62	105.08	5	43.98	59.89
LTA-OL[0.9]	54.13	46.45	100.57	90.65	104.32	13	40.47	64.96
LTA-CS[0.7]	53.79	48.26	102.05	97.45	106.77	5	48.23	55.16
LTA-CS[0.8]	53.65	48.66	102.30	97.47	111.90	5	46.90	57.73
LTA-CS[0.9]	53.50	55.83	109.33	97.96	299.96	0	49.24	59.75

at least in the simple supply chain topology, it makes sense to smoothen the planned order releases through limitation of the workload in future periods but that the actual order releases in the first period should not be restricted in order to avoid idleness.

Overall, the LTA-OL algorithms clearly dominate the other algorithms in terms of average total cost. Although they maintain a higher level of WIP in the PU, these costs are outweighed by the reduction of the safety stock levels. The clearing function with $\varepsilon = 0.2$ performs better than CAP in many cases and CAP performs better than UNC. These experiments indicate that smoothing can reduce the total cost of holding inventory and that taking into account stochasticity may improve the algorithm.

In a number of additional experiments, we study the performance of the planning algorithms for a short planned lead time (set 2) and a long planned lead time (set 3). The results for these experiments are shown in Table 6. For the settings with $L_k = 1$ we see that the performance of all planning algorithms that do not limit the workload (UNC, LTA-OL) in the PU is better than the performance of those that do (CAP, CLF, LTA-CS). The maximum workload that can be processed in a single period according to the anticipation functions is too small to yield a sufficiently high throughput rate. The high safety stocks are put in place to cover the structural deficit in supply of

Table 7 Simulation results for set 4–6

Algorithm	WIP	FGI	Average	Lowest	Highest	Best cases (%)	Sfty as % of total (%)	Fill rate w/o Sfty (%)
Set 4: high utilization (85 %)								
CAP	41.21	58.79	100.00	100.00	100.00	0	65.04	49.01
UNC	46.97	63.25	110.22	103.92	116.89	0	70.02	27.39
CLF[0.05]	41.10	81.44	122.54	91.59	543.46	5	70.07	51.66
CLF[0.2]	37.36	53.09	90.45	78.96	107.82	0	62.95	52.47
LTA-OL[0.7]	41.89	56.15	98.04	90.17	102.59	5	52.95	59.55
LTA-OL[0.8]	41.99	54.25	96.24	89.97	100.18	45	48.55	63.66
LTA-OL[0.9]	41.93	54.26	96.19	90.35	102.62	40	43.15	68.66
LTA-CS[0.7]	40.95	66.08	107.03	95.60	204.48	0	64.46	57.28
LTA-CS[0.8]	40.71	85.99	126.70	96.03	281.90	0	75.62	57.19
LTA-CS[0.9]	40.35	228.73	269.08	96.15	1005.82	5	109.97	54.30
Set 5: low utilization (75)								
CAP	46.54	53.46	100.00	100.00	100.00	0	58.05	52.05
UNC	48.53	57.64	106.17	101.21	113.11	0	62.23	36.34
CLF[0.05]	46.53	53.71	100.24	97.81	102.69	0	57.80	53.22
CLF[0.2]	43.86	50.46	94.32	88.83	98.98	0	57.74	53.19
LTA-OL[0.7]	47.04	53.11	100.15	97.83	103.02	5	53.13	58.88
LTA-OL[0.8]	47.05	51.78	98.83	96.52	100.81	25	50.28	62.57
LTA-OL[0.9]	46.71	51.34	98.05	95.61	100.00	65	46.48	66.93
LTA-CS[0.7]	46.35	54.09	100.44	97.32	102.15	0	54.95	58.19
LTA-CS[0.8]	46.19	53.49	99.68	97.45	102.40	5	52.80	61.26
LTA-CS[0.9]	46.07	54.66	100.73	96.63	107.95	0	50.74	63.79
Set 6: perfect demand forecast								
CAP	62.88	37.12	100.00	100.00	100.00	0	41.07	73.35
UNC	66.71	47.00	113.71	103.68	130.58	0	51.83	32.06
CLF[0.05]	62.80	35.19	97.99	94.26	101.46	0	37.32	77.70
CLF[0.2]	55.38	30.66	86.04	74.88	93.22	0	37.14	77.44
LTA-OL[0.7]	63.42	34.53	97.95	94.44	102.89	0	30.04	83.03
LTA-OL[0.8]	63.34	30.74	94.08	89.34	103.64	0	24.60	86.91
LTA-OL[0.9]	62.84	26.22	89.07	84.71	95.05	75	16.63	91.18
LTA-CS[0.7]	62.61	34.56	97.16	92.09	109.52	0	31.46	81.35
LTA-CS[0.8]	62.36	33.36	95.72	90.74	125.26	0	30.71	81.80
LTA-CS[0.9]	62.08	85.24	147.32	86.02	748.91	25	48.71	82.57

the algorithms where the workload is limited. For the experiments with $L_k = 3$ we see little difference between the algorithms although the CLF[0.2] algorithm seems to perform slightly better than the other algorithms. The small differences in costs between the algorithms can be explained from the smoothing effect of the planned lead time and the pooling of variability in the processing times at higher workloads.

Two sets of simulation experiments (sets 4 and 5) are conducted to study the effect of utilization. The parameters for these experiments are shown in Table 4 and the results are presented in Table 7. These experiments confirm our hypothesis that the benefit of production smoothing is smaller at lower utilization rates. At higher utilization rates, we observe again that those algorithms that restrict the workload require higher safety stocks that lead to higher overall costs.

Finally, we conduct a set of experiments where we study the effect of processing time uncertainty in isolation (set 6) (results in Table 7). In these experiments there is no demand uncertainty. The benefit of the planning algorithms becomes more pronounced in these experiments. Safety stock is still required but is as small as 16.63% of the total costs for LTA-OL with $\phi = 0.9$.

To summarize, we see that an optimistic strategy where overloading is allowed gives better cost performance than a conservative strategy that attempts to maintain lead time feasibility at all times. We furthermore see that algorithms that do not restrict the workload (UNC, LTA-OL) perform best at short planned lead times, and algorithms that do restrict the workload perform better at longer planned lead times. However, at long planned lead times, pooling reduces the variance of supply and requirements and reduce the need to account for the stochastic. As a result, we see smaller differences in the total costs. Overall, the LTA-OL algorithm consistently outperforms the other algorithms. Only in those cases where the planned lead time is long we find that it is sometimes outperformed by the CAP algorithm. Good results are particularly observed for short planned lead times and high utilization levels.

6 Conclusions

The LTA method is a novel approach for modeling the capacity of a PU that is subject to processing uncertainties for SCOP. Contrary to related approaches that model the capacity in a single planning period (e.g. the clearing function), we focus on the capacity over the planned lead time. This allows us to control a network consisting of multi-item PUs without needing to make assumptions on the order of processing. We combine linear programming with a local smoothing heuristic that tracks the stochastic evolution of the workload in the planning horizon and formulate ϕ-percentile constraints rather than constraints on the expected value. Finally, we compare a conservative strategy where the workload is always restricted to an optimistic strategy that allows for temporary overloading if a higher throughput rate is required. In the simulation experiments, the latter strategy proves to be superior in terms of total costs.

The LTA algorithm allows for a investment in inventories that is consistently lower than the investment required for other algorithms under the same service level constraint. The simulation experiments show that the relative cost savings for the LTA algorithm may further be increased if the right planned lead time and value of ϕ are selected. These choices are a topic for further research.

In our simulation experiments we observe that those algorithms that restrict the workload in the PU and do not allow for overloading, perform poorly, particularly if the planned lead time is short. This also explains the mediocre results for

the performance of the clearing function algorithm in some cases. Better performance may possibly be achieved for these algorithms if an overloading option is included. In the presence of the goods flow model, such and extension is not straight forward.

An extension of the work in this paper is to create a more realistic model of the PU. We have represented the PU by its bottleneck resource alone. This simple representation allowed us to use the moment-iteration method to describe the development of the workload over the planning horizon. However, PUs typically have multiple resources that may influence the lead time. It is a challenge to formulate a more comprehensive anticipation function of the PU and find an algorithm that allows it to be optimized in an integral fashion together with the goods flow model.

Another extension of the work presented in this paper is to consider dynamic, workload-dependent planned lead times. The effect of any production smoothing approach with fixed planned lead times is that goods are produced ahead of their requirements. In this paper, we assumed that this additional slack time exists at the SCOP level only and not at the PU level. If it were possible to set deviating planned lead times for some jobs individually (e.g. through a pre-release of jobs), then it is possible to transfer this slack time to the PU allowing for higher workload levels in the PU.

Finally, we mention that it is an important and interesting topic for further research to study the use of safety stock for intermediate items. In this research, the SSAP allowed us to compare costs for different planning algorithms. This procedure places safety stocks only at the final stages of the supply chain. Methods for the placement of safety stocks in general supply chain networks depend to a large extent on reliable constant flow times (see van Houtum 2006). It may well be that the methods presented in this paper facilitate the application of these methods in situations where flow times are random variables that depend on workload.

Appendix A: Identities for the Gamma distribution

Let X be a random variable following a Gamma distributed with parameters α and θ such that

$$\mathbb{E}[X] = \alpha\,\theta, \tag{49}$$

$$\mathrm{Var}[X] = \alpha\,\theta^2 \tag{50}$$

The probability density function of X is

$$g_{\alpha,\theta}(x) = \frac{\theta^{-1}\,\mathrm{e}^{-x/\theta}\left(\frac{x}{\theta}\right)^{\alpha-1}}{\Gamma(\alpha)}, \tag{51}$$

the cumulative distribution function is

$$G_{\alpha,\theta}(y) = \int_{x=0}^{y} \frac{\theta^{-1}\,\mathrm{e}^{-x/\theta}\left(\frac{x}{\theta}\right)^{\alpha-1}}{\Gamma(\alpha)}\,\mathrm{d}x \tag{52}$$

and the complement distribution function is

$$\bar{G}_{\alpha,\theta}(y) = 1 - G_{\alpha,\theta}(y) \tag{53}$$

The first partial moment of the Gamma distribution is

$$\begin{aligned}
\mathbb{E}\left[(X-\lambda)^{+}\right] &= \int_{x=\lambda}^{\infty} (x-\lambda) g_{\alpha,\theta}(x)\mathrm{d}x \\
&= \int_{x=\lambda}^{\infty} (x-\lambda)\frac{\theta^{-1}\,\mathrm{e}^{-x/\theta}\left(\frac{x}{\theta}\right)^{\alpha-1}}{\Gamma(\alpha)}\mathrm{d}x \\
&= \int_{x=\lambda}^{\infty} \frac{\frac{x}{\theta}\,\mathrm{e}^{-x/\theta}\left(\frac{x}{\theta}\right)^{\alpha-1}}{\Gamma(\alpha)}\mathrm{d}x - \lambda\,\bar{G}_{\alpha,\theta}(\lambda) \\
&= \alpha\,\theta \int_{x=\lambda}^{\infty} \frac{\theta^{-1}\,\mathrm{e}^{-x/\theta}\left(\frac{x}{\theta}\right)^{\alpha}}{\Gamma(\alpha+1)}\mathrm{d}x - \lambda\,\bar{G}_{\alpha,\theta}(\lambda) \\
&= \alpha\,\theta\,\bar{G}_{\alpha+1,\theta}(\lambda) - \lambda\,\bar{G}_{\alpha,\theta}(\lambda),
\end{aligned} \tag{54}$$

and the second partial moment is

$$\begin{aligned}
\mathbb{E}\left[\left((X-\lambda)^{+}\right)^2\right] &= \int_{x=\lambda}^{\infty} (x-\lambda)^2 g_{\alpha,\theta}(x)\mathrm{d}x \\
&= \int_{x=\lambda}^{\infty} \frac{x^2\,\theta^{-1}\,\mathrm{e}^{-x/\theta}\left(\frac{x}{\theta}\right)^{\alpha-1}}{\Gamma(\alpha)}\mathrm{d}x \\
&\quad -2\lambda \int_{x=\lambda}^{\infty} \frac{x\,\theta^{-1}\,\mathrm{e}^{-x/\theta}\left(\frac{x}{\theta}\right)^{\alpha-1}}{\Gamma(\alpha)}\mathrm{d}x + \lambda^2\bar{G}_{\alpha,\theta}(\lambda) \\
&= (\alpha+1)\,\alpha\,\theta^2 \int_{x=\lambda}^{\infty} \frac{\theta^{-1}\,\mathrm{e}^{-x/\theta}\left(\frac{x}{\theta}\right)^{\alpha+1}}{\Gamma(\alpha+2)}\mathrm{d}x \\
&\quad -2\lambda\,\alpha\,\theta\,\bar{G}_{\alpha+1,\theta}(\lambda) + \lambda^2\bar{G}_{\alpha,\theta}(\lambda) \\
&= (\alpha+1)\,\alpha\,\theta^2\,\bar{G}_{\alpha+2,\theta}(\lambda) - 2\lambda\,\alpha\,\theta\,\bar{G}_{\alpha+1,\theta}(\lambda) + \lambda^2\bar{G}_{\alpha,\theta}(\lambda),
\end{aligned} \tag{55}$$

These identities are applied to obtain Eqs. (27) and (28). They can also be applied to obtain Eq. (41) noticing that

$$\begin{aligned}\mathbb{E}[\min\{X,\lambda\}] &= \mathbb{E}[\min\{0,-(X-\lambda)\}+X] \\ &= \mathbb{E}[X]-\mathbb{E}\left[(X-\lambda)^{+}\right] \\ &= \alpha\,\theta\,G_{\alpha+1,\theta}-\lambda\,\bar{G}_{\alpha,\theta}(\lambda) \end{aligned} \tag{56}$$

References

Asmundsson J, Rardin RL, Uzsoy R (2006) Tractable nonlinear production planning models for semiconductor wafer fabrication facilities. IEEE Trans Semicond Manuf 19(1):95–111

Asmundsson J, Rardin RL, Turkseven CH, Uzsoy R (2009) Production planning with resources subject to congestion. Naval Res Logist 56(2):142–157. ISSN:1520-6750

Bertrand JWM, Wortmann JC, Wijngaard J (1990) Production control: a structural and design oriented approach. Elsevier, Amsterdam

Bitran GR, Tirupati D (1993) Hierarchical production planning. Handbooks in OR & MS, vol 4, chap 10. North-Holland, pp 523–568

Byrne MD, Bakir MA (1999) Production planning using a hybrid simulation–analytical approach. Int J Prod Econ 59(1–3):305–311

De Kok AG (1989) A moment-iteration method for approximating the waiting-time characteristics of the gi/g/1 queue. Probab Eng Inf Sci 3:273–287

De Kok AG (2003) Evaluation and optimization of strongly ideal assemble-to-order systems. In: Shanthikumar JG, Yao, DD, Zijm WHM (eds) Stochastic modeling and optimization of manufacturing systems and supply chains, pp 203–242

De Kok AG, Fransoo JC (2003) Planning supply chain operations: definition and comparison of planning concepts. In: Handbooks in operations research and management science, vol 11, pp 597–676. ISSN:0927-0507

Graves SC (1986) A tactical planning model for a job shop. Oper Res 34(4):522–533

Hackman ST, Leachman RC (1986) A general framework for modeling production. Manag Sci 35(4): 478–495

Hopp WJ, Spearman ML (2011) Factory physics. Irwin McGraw-Hill

Hung YF, Hou MC (2001) A production planning approach based on iterations of linear programming optimization and flow time prediction. J Chines Inst Ind Eng 18(3):55–67

Hung YF, Leachman RC (1996) A production planning methodology for semiconductor manufacturing-based on iterative simulation and linear programming calculations. IEEE Trans Semicond Manuf 9(2):257–269

Hwang S, Uzsoy R (2005) A single stage multi-product dynamic lot sizing model with work in process and congestion. Technical report, Purdue University

Irdem DF, Kacar NB, Uzsoy R (2008) An experimental study of an iterative simulation-optimization algorithm for production planning. In: Proceedings of the 40th conference on winter simulation, pp 2176–2184

Karmarkar US (1989) Capacity loading and release planning with work-in-progress (WIP) and leadtimes. J Manuf Oper Manag 2:105–123

Karmarkar US (1993) Logistics of production and inventory. In: Manufacturing lead times, order release and capacity loading, vol 4. North-Holland, pp 287–329

Kim B, Kim S (2001) Extended model for a hybrid production planning approach. Int J Prod Econ 73(2):165–173

Kohler-Gudum C, De Kok AG (2002) A safety stock adjustment procedure to enable target service levels in simulation of generic inventory systems. Technical report, Eindhoven University of Technology

Law A (2007) Simulation modeling and analysis, 4th edn. McGraw-Hill

Meyr H, Wagner M, Rohde J (2005) Structure of advanced planning systems. Supply chain management and advanced planning, pp 109–115

Missbauer H (2002) Aggregate order release planning for time-varying demand. Int J Prod Res 40(3): 699–718
Missbauer H (2009) Models of the transient behaviour of production units to optimize the aggregate material flow. Int J Prod Econ 118(2):387–397
Missbauer H (2010) Order release planning with clearing functions: a queueing-theoretical analysis of the clearing function concept. Int J Prod Econ (in press)
Pahl J, Voß S, Woodruff DL (2007) Production planning with load dependent lead times: an update of research. Ann Oper Res 153(1):297–345. ISSN:0254-5330
Riano G (2002) Transient behavior of stochastic networks: application to production planning with load-dependent lead times. PhD thesis, Georgia Institute of Technology
Schneeweiss C (2003) Distributed decision making, 2nd edn. Springer, Berlin
Selçuk B (2007) Dynamic performance of hierarchical planning systems: modeling and evaluation with dynamic planned lead times. PhD thesis, Eindhoven University of Technology
Silver EA, Pyke DF, Peterson R (1998) Inventory management and production planning and scheduling, 3rd edn. Wiley, New York
Simpson NC (1999) Multiple level production planning in rolling horizon assembly environments. Eur J Oper Res 114(1):15–28. ISSN:0377-2217
Spitter JM, Hurkens CAJ, De Kok AG, Lenstra JK, Negenman EG (2004) Linear programming models with planned lead times for supply chain operations planning. Eur J Oper Res 163:706–720
Stadtler H (2005) Supply chain management and advanced planning—basics, overview and challenges. Eur J Oper Res 163(3):575–588
van Houtum GJ (2006) Multiechelon production/inventory systems: optimal policies, heuristics, and algorithms. Tutorial in operations research, INFORMS 2006
Vollmann TE, Berry WL, Whybark DC (1984) Manufacturing planning and control systems. Dow Jones-Irwin, Homewood
Zipkin PH (2000) Foundations of inventory management. McGraw-Hill, New York