A cyclic production scheme for multi-item production systems with backlog : part 1

EURANDOM PREPRINT SERIES
2010-018

A cyclic production scheme for multi-item production systems with backlog; part 1

J. Bruin, J. van der Wal
ISSN 1389-2355

A cyclic production scheme for multi-item production systems with backlog; part 1

J. Bruin*
bruin@eurandom.tue.nl

J. van der Wal†
jan.v.d.wal@tue.nl

May 3, 2010

Abstract

This paper is part 1 of two companion papers dealing with a multi-item production system in which the production is controlled by a fixed cycle scheme. The cycle consists of a production period with a fixed number of production times that can be used for production or idling, followed by a vacation. The duration of the vacation is independent of the production period. Demand arrives according to a (compound) Poisson process and is satisfied from stock or backlogged. The embedded process is modeled in discrete time and analyzed using generating functions. The optimal base stock level is derived from a newsvendor type relation. The model is extended to one with time slot dependent base stock levels. The results are used to construct a presumably optimal fixed cycle policy. In part 2 this fixed cycle policy is used to construct a dynamic production policy.

Keywords: fixed cycle, polling, multi-item production system, discrete time

1 Introduction

In two companion papers we will develop a dynamic control strategy for a multi-item production system, in which for each of the N items, demand arrives according to a (compound) Poisson process and is satisfied from stock or backlogged. In order to satisfy the demand, the machine has to switch between the items which requires a certain switch-over time. A fast switching scheme increases the responsiveness of the system to changes in demand, while a slow switching scheme increases the short term utilization of the system. Decisions on producing, switching and idling the machine should be based on the stock levels of the different product types, with as goal the minimization of a linear cost function in the number of products on stock and the backlog. This is a multi-dimensional decision problem, for which, in principle, the optimal solution can be obtained via MDP. However, this approach becomes numerically intractable if the system is large in the number of product types. Therefore, alternative policies are to be considered.

This multi-item production system can be seen as a polling model, which has been extensively studied in queueing theory with as typical application data transfer in a service switch. Service disciplines that are studied are exhaustive, gated and k-limited control in which respectively all jobs, all jobs upon arrival of the server and at most k jobs are served during a visit, see for example Resing [22]. He gives a moment generating function of the joint queue length distribution at polling instants for so called multitype branching processes, which includes the exhaustive and

*EURANDOM and Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600MB Eindhoven, The Netherlands

†Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600MB Eindhoven, The Netherlands

gated policy in a polling model. The exhaustive, gated and $k-$limited policy can be translated to production strategies by setting base stock levels. According to an exhaustive base stock policy, the machine produces the items in a fixed order and switches to the next item if the current stock level equals the base stock level. In the gated base stock policy, the number of products that is produced during a visit is equal to the number of products short to the base stock level upon arrival. In the $k-$limited base stock policy, the machine switches if the current stock level equals the base stock level or if k units are produced.

Federgruen and Katalan [11] also look at exhaustive and gated base stock policies, but insert a fixed idle time before every switch-over time. Because the total average costs only depend on the total idle time in a cycle, they can derive the optimal idle time via a numerical procedure. The exhaustive, gated and k-limited policies are so-called local lot-sizing policies, where decisions only depend on the stock level or backlog of the product type currently set up. Local lot sizing policies do not react to the complete state of the system and are therefore suboptimal. In Winands et al. [27], where this problem is called a stochastic economic lot sizing problem, more research on dynamic policies is advocated to develop more insights in the optimal policy.

In this first paper we will analyse the system when it is controlled by a so-called fixed cycle. The fixed cycle control means that in each cycle each product receives a fixed number of production slots. These slots can be used for production or for idling, but they cannot be skipped. Given such a fixed cycle, the system decomposes into N independent subsystems, one for each product type. Therefore the analysis of the fixed cycle strategy is much easier than the analysis of practically any other strategy. In the second paper we will use the results for the fixed cycle strategy to construct a dynamic control strategy according to what has become known as the one-step improvement method, which in fact is a single policy improvement step in Howard's policy iteration algorithm. In order to perform such an improvement step one needs the relative values (or bias terms) that in general are too hard to obtain for a system with a very large number of states, but the special structure of the fixed cycle strategy, particularly the fact that it allows for the decomposition, makes computation of these relative values easy.

The idea for this one step improvement approach goes back to Norman [19] and was used in [2], [14] and [23] for call centers, the control of a traffic light and telecommunication systems respectively. Based on the fixed cycle production scheme discussed in this paper, the same approach is performed on a multi-item production system in [4]. The current paper discusses the optimal decision levels for a given fixed cycle and presents a local search algorithm for a (close to) optimal fixed cycle scheme.

Güllü et al. [13] also study the fixed cycle production scheme and present two heuristic algorithms to find the lengths of the production periods of the different items. Both algorithms are (partly) based on a deterministic model. Further, the optimal decision level is derived for a single period model. In Erkip et al. [10], the same model is studied and a matrix analytic method is used to find the optimal decision level for the infinite horizon model. The difference between the model studied in [10] and the model we consider, is that in [10] all time slots have the same length and demand distributions are equal for each time slot.

A more general queueing model with vacations is studied in Fuhrmann and Cooper [12]. They give a decomposition result for the distribution of the number of customers present in a queueing model with vacations which holds under certain conditions. These conditions include: Customers arrive according to a Poisson process, the customers are served in an order that is independent of their service times and the number of customers that arrive during a vacation is independent of the number of customers present just before the start of that vacation. The decomposition consists of the number of customers present in a standard $M|G|1$ queue and the number of customers who arrive during a residual vacation. Unfortunately, the fixed cycle model does not satisfy all necessary conditions, because if the base stock level is reached during a production period, the system idles during one production time. This idling time is also called a vacation and therefore, the number of customers that arrive during a vacation is not independent of the number of customers present in the system when the vacation began. Fortunately, the limiting distribution of the stock out (which can be seen as the queue length distribution of items that need to be produced) can be found in a more direct way, without using the decomposition result of [12].

In this paper, the optimal decision levels for a given fixed cycle are derived from a newsvendor type equation in Section 2 and a heavy traffic approximation is given to avoid numerical problems. Furthermore, a fixed cycle scheme with time slot dependent decision levels is analysed in Section 3 and again a newsvendor type equation is given. Section 4 presents an algorithm to find a near-optimal fixed cycle scheme. A summary is given in Section 5.

2 Fixed Cycle

2.1 Cyclic production

The fixed cycle policy reserves a production period of fixed length for every item i. This production period consists of a number of g_i production times, each with a length $T_{P,i}$. The order of production is fixed and the decision to produce or not to produce a product is based on the base-stock level S_i. If the stock level of item i equals this level just before a production slot of type i, the system idles during the next slot. This clearly is suboptimal, but this condition (the fact that slots aren't skipped) allows us to analyse the system as a combination of N *independent* queues, one for each product type. Every queue can be seen and analysed as a single-item production system with periodic vacations, which is done in the first part of this paper. Because the analysis is focussed on only one of the N product types, the index i is omitted in the notation.

So, we consider a single-item cyclic production system where each cycle starts with g production times and is concluded by a vacation. This vacation period consists of the reserved production periods for the other items and the total time spent on switching, say σ. Production and vacation times are possibly random but independent. Demand arrives according to a (compound) Poisson process. The system is embedded on the instances corresponding to the start of a production time or the start of a vacation. The time intervals in this chain will be called *slots*, where each cycle consists of g production slots and 1 vacation slot. Whether a production slot is used for production or for idling is read from a base stock level S. If at the start of a production slot the stock level is less than S, then the slot is used to produce exactly one item. Given the assumptions about the demand process and the production rule we obtain a periodic (cyclic) Markov chain embedded at the moments a slot starts.

The state of this chain will be described by the number of products short to the level S at the beginning of a slot and the slot number within the cycle. Using this formulation, the limiting behavior of the Markov chain is independent of the value of S. Linear cost functions are considered for the number of items on stock and the backlog. Then, as we will show, if the distribution of the number of products short to the base stock level S is known, an expression for the optimal value S^* can be derived from a newsvendor type equation. This stock-out distribution will be determined via a generating function approach.

To this end, first some notation is introduced. The generating function for the demand in a production slot is denoted by $\mathcal{A}_P(z) = \sum_{k=0}^{\infty} a_P(k)z^k$, with $a_P(k)$ the probability that the demand in a production slot is equal to k. The average length of a production slot is denoted by T_P. Similarly, $\mathcal{A}_V(z)$, $a_V(k)$ and T_V are defined for the vacation slot. Further, λ denotes the mean demand per time unit and X is defined as the number of products short to the level S. In the following subsection an expression is derived for $G_n(z)$, the generating function of X at slot boundary n.

2.2 The generating function

Define $X_{n,t}$ as the value of X at slot boundary n in cycle t for $n = 1, \ldots, g+1$. Now consider the limiting random variable

$$X_n = \lim_{t \to \infty} X_{n,t}, \quad n = 1, \ldots, g+1.$$

Denote its limiting distribution by

$$p(k, n) = \mathbb{P}(X_n = k), \quad k \geq 0, \; n = 1, \ldots, g+1$$

and the generating function of X_n by

$$\mathcal{G}_n(z) = \sum_{k=0}^{\infty} p(k, n) z^k, \quad n = 1, \ldots, g+1.$$

The distribution of X_n is well-defined if the Markov chain $\{X_{n,t}, t = 1, 2, \ldots\}$ is aperiodic and irreducible (which is immediate from the (compound) Poisson demand assumption) and provided the system is stable, i.e. if the number of arrivals per cycle is less than the available number of production slots, so if $\lambda\,(gT_P + T_V) < g$. We assume this is the case.

Then, with A_n the demand that occurs in time slot n

$$\begin{aligned} X_1 &= X_{g+1} + A_{g+1}, \\ X_n &= X_{n-1} + A_{n-1} - I_{\{X_{n-1} > 0\}}, \quad n = 2, \ldots, g+1. \end{aligned}$$

From these equations one gets

$$\begin{aligned} \mathcal{G}_1(z) &= \mathcal{G}_{g+1}(z) \mathcal{A}_V(z), \\ \mathcal{G}_n(z) &= \frac{1}{z} \mathcal{A}_P(z) \left[\mathcal{G}_{n-1}(z) + p(0, n-1)(z-1) \right], \quad n = 2, \ldots, g+1. \end{aligned}$$

As one easily verifies, this leads by iteration to

$$\mathcal{G}_1(z) = \frac{\sum_{m=1}^{g} \mathcal{A}_P^{g+1-m}(z) \mathcal{A}_V(z) (z^m - z^{m-1}) p(0, m)}{z^g - \mathcal{A}_P^g(z) \mathcal{A}_V(z)}, \tag{1}$$

$$\mathcal{G}_n(z) = \left(\frac{\mathcal{A}_P(z)}{z} \right)^{n-1} \mathcal{G}_1(z) + \sum_{m=1}^{n-1} p(0, m)(z-1) \left(\frac{\mathcal{A}_P(z)}{z} \right)^{n-m}, \quad n = 2, \ldots, g+1. \tag{2}$$

The generating function of X_1 is of indeterminate form, but the g boundary probabilities $p(0, n), \; n = 1, \ldots, g$, can be determined by considering the zeros of the denominator in (1) that lie on or within the unit circle. The following Rouché type lemma is taken from [1] and is specialized to our case.

Lemma 1 *If $\rho := \frac{\lambda(gT_P + T_V)}{g} < 1$ and $\mathcal{A}_V(0)\mathcal{A}_P^g(0) \neq 0$, then $z^g = \mathcal{A}_P^g(z)\mathcal{A}_V(z)$ has g roots on or within the unit circle.*

Denote the g roots of $z^g = \mathcal{A}^{g+1}(z)$ in $|z| \leq 1$ by $z_0 = 1, z_1, \ldots, z_{g-1}$. Since the function $\mathcal{G}_1(z)$ is finite on and inside the unit circle, the numerator of the right-hand side of (1) needs to be zero for each of the g roots, i.e., the numerator should vanish at the exact points where the denominator of the right-hand side of (1) vanishes. Lemma 1 and (1) together lead to g equations in terms of the g boundary probabilities, from which the latter can be determined. The roots can be determined using methods from [16]. It is assumed that these g roots are all different. If roots with multiplicity greater than one occur, the derivatives (up to the number of multiplicity) of the numerator of (1) can be set to zero to obtain sufficiently many equations. For the root $z = 1$, l'Hôpital's rule is applied to obtain one equation from $G_1(1) = 1$.

2.3 The limiting distribution

In principle, the probabilities $p(k,n)$ can be found by numerically inverting $\mathcal{G}_n(z)$. However, in this case, the probabilities can be derived directly from the g boundary probabilities.

In order to find all limiting probabilities $p(k,n)$ from the probabilities $p(0,n),\ n = 1,\dots,g$ (obtained via Lemma 1), the balance equations are used:

$$p(k,1) = \sum_{j=0}^{k} p(j,g+1)a_v(k-j), \tag{3}$$

$$p(k,n) = \sum_{j=1}^{k+1} p(j,n-1)a_p(k+1-j) + p(0,n-1)a_p(k),\ n = 2,\dots,g+1. \tag{4}$$

Using (3) with $k = 0$, the probability $p(0,g+1)$ can be obtained from

$$p(0,g+1) = \frac{1}{a_v(0)}p(0,1). \tag{5}$$

Next let us rewrite equation (4) for $n = 2,\dots,g+1$ as

$$p(k+1,n-1) = \frac{1}{a_p(0)}\left(p(k,n) - \sum_{j=1}^{k} p(j,n-1)a_p(k+1-j) - p(0,n-1)a_p(k)\right). \tag{6}$$

Then, starting with $k = 0$, we first find the probabilities $p(1,n),\ n = 2,\dots,g+1$. The probability $p(1,1)$ can then be obtained from equation (3). Continuing in this way, one recursively gets the probabilities $p(k,n),\ k \geq 2$.

2.4 The optimal base-stock level

In this section the optimal base stock level will be determined for the case of linear holding and backlogging costs. These costs will be computed from the expected number of products on stock or backlogged at slot boundaries. Therefore, the vacation period is divided into a number of vacation slots, say g_V, such that the slots are small enough to get a good approximation for the expected costs per time unit. The lengths of the vacation slots are all equal to $T_v := \frac{T_V}{g_V}$. Because the demand distribution is assumed to be (compound) Poisson, the stock-out distribution at the new slot boundaries can easily be found, using the stock-out distribution at slot boundary g (which was already found in the previous section).

The expected stock-out at slot boundary $g+n, n = 1,\dots,g_V$ is just $\mathbb{E}(X_g) + n\lambda T_v$. Now define the following linear cost function, with weights based on the average slot duration:

$$\begin{aligned} C(S) &= \sum_{n=1}^{g} \frac{T_P}{gT_P + T_V}\left(c_I\mathbb{E}(I(n)) + c_B\mathbb{E}(B(n))\right) \\ &+ \sum_{n=g+1}^{g+g_V} \frac{T_v}{gT_P + T_V}\left(c_I\mathbb{E}(I(n)) + c_B\mathbb{E}(B(n))\right), \end{aligned} \tag{7}$$

where $I(n)$ is the number of items on stock and $B(n)$ the backlog at slot boundary n. Because the cost function is a weighted sum of costs at different time slots, we also look at the corresponding weighted limiting distribution:

$$p(k)=\sum_{n=1}^{g}\frac{T_P}{gT_P+T_V}p(k,n)+\sum_{n=g+1}^{g+g_V}\frac{T_v}{gT_P+T_V}p(k,n)\ .$$

The optimal base-stock level S^* for this 'newsvendor problem' (see for example Porteus [21]) is now readily obtained as:

$$S^*=\min\left\{S\left|\sum_{k=0}^{S}p(k)>\frac{c_B}{c_I+c_B}\right.\right\}. \tag{8}$$

2.5 A geometric tail approximation

The probabilities $p(k)$ in (8) can be found using the recursive method from Section 2.3. However, we have experienced numerical problems with this procedure for large values of k. If the load on the system is high and S gets larger than 20, the recursive method gives numerically unstable results. Therefore we propose to use the following approximation for $p(k)$ if S^* gets large.

In [9], Van Eenige encounters the same numerical problems and uses an approximation from Tijms and Van de Coevering [25] for the tail probabilities that is based on the following asymptotic behavior

$$\lim_{k\to\infty}\frac{p(k)}{p(k+1)}=\gamma,$$

with γ the unique root of $z^g-\mathcal{A}_P^g(z)\mathcal{A}_V(z)$ in $(1,\infty)$. This root can easily be computed with bisection.

Let us use the direct computation of p_k up to K and the tail approximation for $k>K$. (The choice of K can be made during the direct computation. If either the geometric behavior seems to have started or one seems to lose the numerical stability, one switches to the geometric tail behavior.)

So, we use

$$\mathbb{P}(X=k)\approx\kappa\gamma^{-k}, k=K+1,\ldots, \tag{9}$$

where κ is the normalization constant which can be expressed in terms of $P(X\le K)$:

$$\kappa=\left(1-\gamma^{-1}\right)\left(1-P(X\le K)\right)\gamma^{K+1}.$$

Upon substituting (9) into (8), and assuming that $S^*>K$, so that the tail approximation holds, one gets the following approximative value for S^*:

$$\tilde{S}=\left\lceil\frac{-\ln(c_I)+\ln(c_I+c_B)+\ln(\kappa)-\ln(\gamma-1)}{\ln(\gamma)}\right\rceil, \tag{10}$$

with $\lceil x\rceil$ the smallest integer that is greater than or equal to x.

In order to see whether (10) results in a good approximation, we give some numerical results comparing the approximation with the exact method.

2.6 Numerical results

For various parameters settings, for which we can determine the exact value of S^* numerically, the results for S^* and $\tilde{S}$ are presented in Tables 1, 2 and 3. These results are based on a fixed cycle scheme with deterministic time slots of unit length and a Poisson demand process. The values of $\tilde{S}$ from (10) are given without taking the 'ceiling' to show the real difference with the value of S^*.

$c_I = 1, c_B = 10, g = 5, T_P = 1, T_V = 5$

ρ	$\mathbb{E}I$	$\mathbb{E}B$	S^*	costs	$\tilde{S}$	costs
0.50	1.26	0.12	2	2.50	2.12	2.58
0.60	1.92	0.11	3	2.98	2.65	2.98
0.70	2.48	0.14	4	3.89	3.59	4.89
0.80	3.65	0.21	6	5.78	5.54	5.78
0.90	7.32	0.44	12	11.68	11.48	11.68
0.95	14.74	0.89	24	23.62	23.45	23.62

Table 1: The values of $\mathbb{E}I$, $\mathbb{E}B$, S^* and $\tilde{S}$.

$c_I = 1, c_B = 10, g = 10, T_P = 1, T_V = 10$

ρ	$\mathbb{E}I$	$\mathbb{E}B$	S^*	costs	$\tilde{S}$	costs
0.50	1.89	0.14	3	3.30	3.69	3.37
0.60	2.48	0.13	4	3.81	4.07	4.00
0.70	2.96	0.17	5	4.62	4.85	4.62
0.80	4.04	0.22	7	6.27	6.63	6.27
0.90	7.62	0.43	13	11.91	12.40	11.91
0.95	15.00	0.88	25	23.71	24.29	23.71

Table 2: The values of $\mathbb{E}I$, $\mathbb{E}B$, S^* and $\tilde{S}$.

One sees that the approximation $\tilde{S}$ is correct for nearly all parameter settings, except for $\rho = 0.5$ in the first table and $\rho = 0.5, 0.6$ in the second table. For higher values of ρ, the approximation is equal to S^*, which is just what we want, because the numerical problems occur if ρ is high. Further, it is observed that the approximation of S^* is less accurate if S^* is low, because it is based on (9), which is only an approximation for the tail probabilities. But for low values of S^*, there are no numerical problems, so this is (again) not a problem.

Figure 1 gives the effect of g on the value of S^* and shows that S^* increases if g decreases. This is explained by the fact that decreasing g increases the effective utilization. Another point is that if g is somewhat longer, one needs the safety stock for the vacation period only at the end of the production period. This suggests that a cost reduction can be obtained with a base-stock level that is lower at the beginning of a production period and increases towards the end of the production period.

$c_I = 1, c_B = 10, g = 3, T_P = 1, T_V = 9$

ρ	$\mathbb{E}I$	$\mathbb{E}B$	S^*	costs	$\tilde{S}$	costs
0.50	1.31	0.10	2	2.29	1.86	2.29
0.60	2.00	0.09	3	2.85	2.42	2.85
0.70	2.58	0.12	4	3.79	3.37	3.79
0.80	3.78	0.19	6	5.71	5.33	5.71
0.90	7.46	0.42	12	11.66	11.29	11.66
0.95	14.89	0.87	24	23.57	23.26	23.57

Table 3: The values of $\mathbb{E}I$, $\mathbb{E}B$, S^* and $\tilde{S}$.

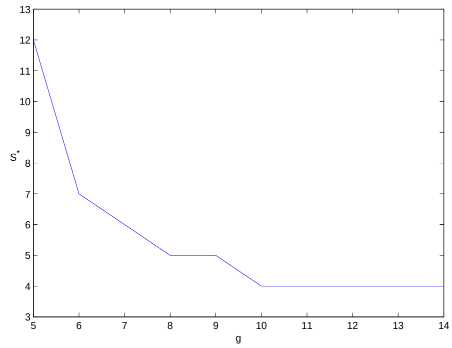

Figure 1: The optimal base stock level, for different lengths of the production period, $T_P = 1, T_V = 4, \lambda = 0.5, c_I = 1, c_B = 10$.

3 Time slot dependent base-stock levels

So, let us now consider the system in which the base-stock levels are time slot dependent. Denote the different base-stock levels by $S_1, \ldots, S_g$, with S_n the base-stock level for time slot n, see Figure 2. In the sequel, particularly the analysis in Subsection 3.1, we will often use the following two assumptions:

Assumption 1

$$S_n \quad \leq \quad S_{n+1},\ n = 1, \ldots, g.$$

Assumption 2

$$S_{n+1} \quad \leq \quad S_n + 1,\ n = 1, \ldots, g.$$

For most realistic settings, these assumptions will hold, but it might be possible to construct counterexamples for this based on the following intuition.

Assumption 1 might be violated in the following situation. If the production times are highly variable and the non-production slots are not, then one might need a higher safety stock at the beginning of the production period than near the end of it.

With respect to Assumption 2 the following could occur. At the end of the production period one wants a higher base stock level because of the coming long vacation period. However, because of the holding costs, one does not want to invest in this higher stock in the production slots before the final one. However, if less than expected demand arrived during the last cycle and there are still $S_{g-1}+1$ products on stock at the start of the $g-th$ production slot, then one might be willing to produce one more product, which would mean $S_g > S_{g-1} + 1$.

In order to find the optimal values of $S_1, \ldots, S_g$, we slightly adapt the model description from Section 2.1. With different base-stock levels (and compound Poisson demand) in every slot the stock level can reach the maximum of $S_1, \ldots, S_g$, so in some slot(s) n the actual stock can be larger than the base-stock level S_n.

This is shown in Figure 3, where the stock level just before the first production slot in the second cycle is higher than S_1.

Therefore, define $S^{\max}$ as $\max\{S_1, \ldots, S_g\}$ and let $\tilde{X}_n$ denote the number of products short compared to $S^{\max}$ at slot boundary n.

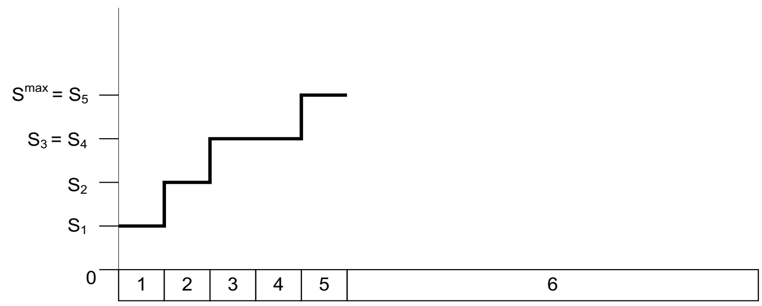

Figure 2: The base stock levels of 5 production slots.

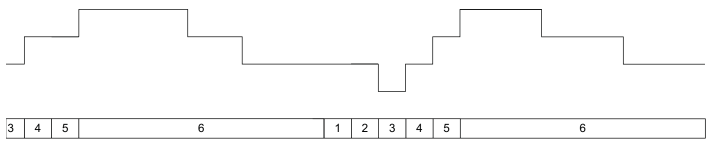

Figure 3: The inventory level during two cycles.

As before, if $\rho < 1$ the limiting distribution of $\tilde{X}_n$ exists and it will be denoted by

$$\tilde{p}(k,n) = \lim_{t\to\infty} \mathbb{P}(\tilde{X}_{n,t} = k), \quad n = 1,\dots,g+1, \quad k \geq 0,$$

with generating function

$$\tilde{\mathcal{G}}_n(z) = \sum_{k=0}^{\infty} \tilde{p}(k,n) z^k, \quad n = 1,\dots,g+1.$$

Define

$$\delta(S_n) := S^{\max} - S_n, \quad n = 1,\dots,g.$$

In the same way as in Section 2.2, we get

$$\begin{aligned} \tilde{\mathcal{G}}_1(z) &= \frac{\sum_{m=1}^{g} \sum_{k=0}^{\delta(S_m)} \tilde{p}(k,m)(z^{k+m} - z^{k+m-1}) \mathcal{A}_P^{g+1-m}(z) \mathcal{A}_V(z)}{z^g - \mathcal{A}_P^g(z)\mathcal{A}_V(z)}, \qquad (11) \\ \tilde{\mathcal{G}}_n(z) &= \tilde{\mathcal{G}}_1(z) \left(\frac{\mathcal{A}_P(z)}{z}\right)^{n-j} + \sum_{m=1}^{n-1} \sum_{k=0}^{\delta(S_m)} \tilde{p}(k,m)(z^k - z^{k-1}) \left(\frac{\mathcal{A}_P(z)}{z}\right)^{n-m}, \\ & \quad n = 2,\dots,g+1, \end{aligned}$$

The expressions for $\tilde{\mathcal{G}}_1(z),\dots,\tilde{\mathcal{G}}_{g+1}(z)$, however, still contain the unknown boundary probabilities $\tilde{p}(k,n),\ k = 0,\dots,\delta(S_n),\ n = 1,\dots,g$. Lemma 1 gives g equations. Since there are more than g unknowns, we will have to construct a larger set of balance equations for these boundary probabilities. A similar problem is discussed in [8]. In the next subsection we will follow the approach used there to find these boundary probabilities.

3.1 The boundary probabilities

The boundary probabilities we are looking for only concern probabilities from the production period. For ease of notation we combine the last production slot and the vacation period into one production slot.

With $a^*_g(k),\ k \geq 0$, denoting the distribution of the total demand in time slots g and $g+1$ together, the set of balance equations becomes:

$$
\begin{aligned}
\tilde{p}(k,n) &= \sum_{m=\delta(S_{n-1})+1}^{k+1} \tilde{p}(m,n-1)a_p(k+1-m) + \sum_{m=0}^{\delta(S_{n-1})} \tilde{p}(m,n-1)a_p(k-m), \\
&\quad 2 \leq n \leq g, \quad k \geq \delta(S_{n-1}), && (12) \\
\tilde{p}(k,n) &= \sum_{m=0}^{k} \tilde{p}(m,n-1)a_p(k-m), \\
&\quad 2 \leq n \leq g, \quad 0 \leq k < \delta(S_{n-1}), && (13) \\
\tilde{p}(k,1) &= \sum_{m=\delta(S_g)+1}^{k+1} \tilde{p}(m,g)a^*_g(k+1-m) + \sum_{m=0}^{\delta(S_g)} \tilde{p}(m,g)a^*_g(k-m), \\
&\quad k \geq \delta(S_g), && (14) \\
\tilde{p}(k,1) &= \sum_{m=0}^{k} \tilde{p}(m,g)a^*_g(k-m), \quad 0 \leq k < \delta(S_g). && (15)
\end{aligned}
$$

Under the assumption that $S_n \leq S_{n-1} + 1$ for all n (see Assumption 2), it would be enough to look at the equations described by (13), (15) and the equations from Lemma 1. However, if this assumption does not hold, then for one or more equations described by (13) and (15) the probability on the left hand side does not appear in the expression for $G_1(z)$. Therefore, below an algorithm is given that results in a set of balance equations which, combined with the equations from Lemma 1 gives us the boundary probabilities that appear in (1).

The algorithm below uses two sets: a set of unknown probabilities (variables), U, and a set of equations, E, from which the unknowns have to be obtained. Initially we define $U = \{\tilde{p}(k,n); k = 0,\ldots,\delta(S_n),\ n = 1,\ldots,g\}$ and we let E contain the g equations from Lemma 1 plus the equations described by (13) and (15). The balance equation with left-hand side $\tilde{p}(k,n)$ will be labeled with (k,n). So at the start U contains the variables $\tilde{p}(k,n), k = 0,\ldots,\delta(S_n)$, and E the equations $(k,n),\ k = 0,\ldots,\delta(S_{n-1}) - 1$. Then the number of equations in E (including the ones from Lemma 1) and the number of unknowns in U are both equal to $\sum_n \delta(S_n) + g$. However, not all the probabilities that appear in the equations in E are in U. For each of these probabilities, an equation will be added to E.

We start with the slot *just after* the one with the lowest decision level, thus the largest $\delta(S_n)$. Let n be the current production slot. For each $(k,n) \in E$ for which $\tilde{p}(k,n)$ is not yet in U the variable $\tilde{p}(k,n)$ is added to U.

Next, for each of these variables an extra balance equation is added to E, namely equation $(k-1,n+1)$ (where $g+1$ is to be read as 1). All probabilities appearing at the right-hand side of this new equation are already in U and at most one extra unknown probability appears at the left-hand side.

Then we move to the next slot, $n+1$. Again each variable that appears in E but is not in U is added to U, and in the same way as before, for a new variable $\tilde{p}(k,n)$ equation $(k-1,n+1)$ is added to E. Continue until all slots have been considered. In the last step (step g), the slot with the highest value of $\delta(S_n)$ is reached. Therefore, the probabilities on the left-hand side of all equations added in the previous step are already in U, because k can not exceed $\max\{\delta(S_1),\ldots,\delta(S_g)\}$.) This means that the construction ends with $|E| = |U|$ and the variables in U being the only ones appearing in E.

Assuming that all equations from the roots, from l'Hôpital's rule, and obtained via this algorithm are linearly independent, the unknowns in U can be found.

3.2 Optimal value for $S^{\max}$

Denote the number of products short compared to $S^{\max}$ at 'weighted' random slot boundaries by $\tilde{X}$. The generating function of $\tilde{X}$ is defined as

$$\tilde{\mathcal{G}}(z) = \sum_{k=0}^{\infty} \tilde{p}(k) z^k = \sum_{n=1}^{g} \frac{T_P}{gT_P + T_V} \mathcal{G}_n(z) + \frac{T_V}{gT_P + T_V} \mathcal{G}_{g+1}(z),$$

with

$$\tilde{p}(k) = \sum_{n=1}^{g} \frac{T_P}{gT_P + T_V} p(k,n) + \frac{T_V}{gT_P + T_V} p(k, g+1), \quad k \geq 0.$$

The limiting distributions of $\tilde{X}, \tilde{X}_1, \ldots, \tilde{X}_g$ can be found by inverting $\tilde{G}(z)$ and $\tilde{\mathcal{G}}_n(z),\ n = 1, \ldots, g$. The distribution of $\tilde{X}$ depends on $\delta(S_1), \ldots, \delta(S_g)$, but not on $S^{\max}$. Therefore, a newsvendor type equation can be given for the optimal value of $S^{\max}$. For a given vector $(\delta(S_1), \ldots, \delta(S_g))$, the optimal value of $S^{\max}$ is given by

$$S^{\max *} = \min \left\{ S^{\max} \left| \sum_{k=0}^{S^{\max}} \tilde{p}(k) > \frac{c_B}{c_I + c_B} \right. \right\}. \tag{16}$$

We emphasize that the distribution of $\tilde{X}$ depends on the whole vector $(\delta(S_1), \ldots, \delta(S_g))$ and thus $S^{\max *}$ does as well. Furthermore, there is no expression for the optimal value of every individual $S_n,\ n = 1, \ldots, g$.

In order to also find the expected costs $C(S_1, \ldots, S_g)$ for a given vector $\{S_1, \ldots, S_g\}$, one can write

$$C(S_1, \ldots, S_g) = (c_I + c_B) \sum_{k=0}^{S^{\max}} \tilde{p}(k)(S^{\max} - k) + c_B(\mathbb{E}\tilde{X} - S^{\max}), \tag{17}$$

where the weights $\frac{T_P}{gT_P+T_V}$ and $\frac{G_v}{gT_P+T_V}$ are now hidden in $\tilde{p}(k)$ and $\mathbb{E}\tilde{X}$.

The finite sum $\sum_{k=0}^{S^{\max}} \tilde{p}(k)(S^{\max} - k)$ is obtained from the equilibrium probabilities. For $\mathbb{E}\tilde{X}$ we use $\mathbb{E}(\tilde{X}_1), \ldots, \mathbb{E}(\tilde{X}_{g+1})$. For the derivation of $\mathbb{E}(\tilde{X}_1)$, we refer to Appendix A. The expressions for $\mathbb{E}(\tilde{X}_n),\ n = 2, \ldots, g+1$ are then obtained with

$$\tilde{X}_n \quad = \quad \tilde{X}_{n-1} + A_{n-1} - I_{\tilde{X}_{n-1} > \delta(S_{n-1})}, \quad n = 2, \ldots, g+1,$$

with A_{n-1} the number of arrivals in time slot $n-1$ and $I_{\tilde{X}_{n-1} > \delta(S_{n-1})}$ the production indicator of time slot $n-1$. The result is:

$$\begin{aligned}
\mathbb{E}(\tilde{X}_1) &= \frac{1}{g - \lambda\,(gT_P + T_V)} \left(\sum_{m=1}^{g} \sum_{k=0}^{\delta(S_m)} \tilde{p}(k,m) \left[(gT_P + T_V)\,\lambda + k + m - 1 \right] \right. \\
&\quad \left. - \frac{1}{2} \left[g(g-1) - (gT_P + T_V)^2 \lambda^2 \right] \right), \\
\mathbb{E}(\tilde{X}_n) &= \mathbb{E}(\tilde{X}_1) + \lambda\,(gT_P + T_V) - \sum_{m=1}^{n-1} \sum_{k=0}^{\delta(S_m)} \tilde{p}(k,m), \quad n = 2, \ldots, g+1.
\end{aligned}$$

3.3 Numerical results

Some numerical results are presented to compare the expected costs per time unit for fixed cycles with a constant decision level with the costs for fixed cycles with time slot dependent decision levels.

Recall that given the vector $(\delta(S_1), \ldots, \delta(S_g))$ the distribution of $\tilde{X}$ does not depend on $S^{\max}$ and that the corresponding optimal $S^{\max}$ is given by (16).

In order to limit the number of possible vectors δ Assumptions 1 and 2 are used. The number of different values for δ then equals 2^{g-1}. In the numerical results below, the presented optimal values $\{S_1, \ldots, S_g\}$ are the optimal ones given these two restrictions.

Table 4:
$c_I = 1, c_B = 10, g = 5, T_P = 1, T_V = 5$

ρ	S^*	costs	$[S_1, S_2, S_3, S_4, S_5]$	costs	cost reduction in %
0.75	5	4.642	[4 4 5 5 5]	4.608	0.716
0.8	6	5.782	[5 5 6 6 6]	5.762	0.334
0.85	8	7.731	[7 7 8 8 8]	7.716	0.195
0.9	12	11.682	[11 11 12 12 12]	11.672	0.089
0.95	24	23.630	[23 23 24 24 24]	23.625	0.023

Table 5:
$c_I = 1, c_B = 20, g = 5, T_P = 1, T_V = 5$

ρ	S^*	costs	$[S_1, S_2, S_3, S_4, S_5]$	costs	cost reduction in %
0.75	6	5.807	[4 5 6 6 6]	5.699	1.869
0.8	8	7.262	[7 7 8 8 8]	7.239	0.324
0.85	10	9.794	[9 9 10 10 10]	9.769	0.248
0.9	15	14.749	[15 15 15 15 15]	14.749	0.000
0.95	30	30.000	[29 30 30 30 30]	29.963	0.110

Table 6:
$c_I = 1, c_B = 10, g = 5, T_P = 1, T_V = 25$

ρ	S^*	costs	$[S_1, S_2, S_3, S_4, S_5]$	costs	cost reduction in %
0.75	5	4.984	[3 4 5 5 6]	4.933	1.024
0.8	7	6.039	[4 5 6 7 7]	6.006	0.551
0.85	9	7.931	[6 7 8 8 9]	7.902	0.365
0.9	12	11.816	[10 11 11 12 13]	11.798	0.150
0.95	24	23.692	[22 23 23 24 24]	23.688	0.015

Tables 4 and 5 show numerical results for $T_V = 5$, while in Tables 6 and 7, $T_V = 25$. In Tables 4 and 6 we have the results for $c_I = 1, c_B = 10$, while in Tables 5 and 7 $c_B = 20$. For almost every value of $\rho < 0.95$, Tables 6 and 7 show a larger cost reduction than Tables 4 and 5. Apparently, the length of the vacation period has a positive effect on the attainable cost reduction, while the value of ρ and the fraction $\frac{c_I}{c_I + c_B}$ have a negative effect on it. The first observation can be explained by the fact that the average number of arrivals per time slot decreases if T_V increases, so there is relatively more time to get a high stock level at the end of the production period. Therefore, the decision levels at the beginning of the production period can be lower. The second observation is easy to explain: The values of ρ increase the load on the system, so the system should use its

Table 7:
$c_I = 1, c_B = 20, g = 5, T_P = 1, T_V = 25$

ρ	S^*	costs	$[S_1, S_2, S_3, S_4, S_5]$	costs	cost reduction in %
0.75	7	6.118	[4 5 6 6 7]	5.976	2.329
0.8	8	7.448	[6 6 7 7 8]	7.425	0.309
0.85	11	9.961	[9 9 10 11 11]	9.894	0.670
0.9	16	14.946	[14 14 15 16 16]	14.856	0.604
0.95	31	30.029	[29 30 30 31 31]	29.987	0.140

full capacity to get $S^{\max*}$ products on stock. The last observation is explained by the fact that if the backlogging costs are relatively high, one wants to prevent the system to create backlog. Therefore, the decision levels are all close to $S^{\max*}$.

The optimal values of δ are more difficult to find than $S^{\max*}$. The main problem here is that the number of possible vectors is too large. Therefore, time slot dependent base stock levels will not be discussed in more detail.

4 The production periods

The (close to) optimal lengths of the production periods can be determined with the local search algorithm presented in Subsection 4.2. Although this algorithm can deal with both constant and time slot dependent base stock levels, we will focus on constant base stock levels. (As we have seen, the search for an optimal vector of base stock levels is time consuming and the difference in costs with the constant base stock level is limited.)

In the remainder of the paper, a fixed cycle will be described by $g = (g_1, \ldots, g_N)$, a vector with the lengths of the production periods. With $T_{P,i}$ denoting the average length of a production slot of item i, the average duration of the cycle, denoted as $T_{FC}(g)$, satisfies $T_{FC}(g) = \sum_i g_i T_{P,i} + \sigma$. Such a cycle is stable if for every item the number of production slots in it suffices, i.e, if $\lambda_i T_{FC}(g) < g_i$ for all i.

4.1 The shortest stable fixed cycle

The local search algorithm presented in Subsection 4.2 starts with a stable cycle of minimum length.

Lemma 2 *If $\rho = \sum_{i=1}^{N} \lambda_i T_{P,i} < 1$, there exists a unique stable fixed cycle of minimum length.*

A proof of this lemma is found in Appendix B.

This cycle will be referred to as the shortest stable fixed cycle and will be denoted by g^{min}, with $T_{FC}^{min} = \sum_{i=1}^{N} g_i^{min} T_{P,i} + \sigma$ the length of this cycle. The next algorithm produces this shortest fixed cycle.

Algorithm 1

Step 1: *Set $n = 0$ and $g^{(0)} = \{1, 1, \ldots, 1\}$ (or alternatively $g_i^{(0)} = \lambda_i \frac{\sigma}{1-\rho}$ for all i, see Appendix C, $g^{(0)} \leq g^{min}$.*

Step 2: *Compute $T_{FC}(g^{(n)}) = \sum_{i=1}^{N} g_i^{(n)} T_{P,i} + \sigma$.*
If the system is stable, i.e., $\lambda_i T_{FC}(g^{(n)}) < g_i^{(n)}$ for all i, then the minimal fixed cycle has been found: $g^{min} = g^n$. Otherwise go to Step 3.

Step 3: *Compute:* $g_i^{(n+1)} = \left\lfloor \lambda_i \left(\sum_{i=1}^{N} g_i^{(n)} T_{P,i} + \sigma \right) + 1 \right\rfloor$, *with* $\lfloor x \rfloor$ *the largest integer less than or equal to* x.
Set $n = n + 1$ *and go back to Step 2.*

Lemma 3 *Algorithm 1 gives the shortest stable fixed cycle for any polling system with* N *queues.*

A proof of this lemma is found in Appendix C.

4.2 A local search algorithm for a good fixed cycle

In order to find a good fixed cycle, we will start with g^{min} and apply a local search algorithm, which works as follows. In each cycle improvement step we lengthen the cycle for one of the product types, the one for which lengthening given the largest reduction in costs. However, there will be two complicating factors. First of all, lengthening the cycle for one type might result in an unstable system for one or more of the other types; this will be solved by lengthening the production period for those product types as well. And second, it is possible that a longer production period for only one product gives an increase in costs, whereas a longer production period for two or more products gives a decrease in costs. This will be taken care of using a special termination criterion; stop only if for a number of improvement steps no improvement has been found. This number is chosen equal to N, so that for each product type simultaneously a production slot can be added.

Algorithm 2

Step 1: *Start with the shortest stable fixed cycle that can be obtained with Algorithm 1 and define* $g^{(1,0)} = g^{min}$.

Step 2: *In improvement step* n, *starting with the cycle* $g^{(n,0)}$ *try lengthening the production period of every product type.*
For item i, *add a production slot for this type obtaining the cycle* $g^{(n,0)} + e_i$, *make this cycle stable, denote this stable cycle by* $g^{(n,i)}$ *and calculate the expected costs* $C(g^{(n,i)})$.
Determine i^* *such that* $C(g^{(n,i^*)}) = \min_i C(g^{(n,i)})$ *and define* $g^{(n+1,0)} = g^{(n,i^*)}$.
Set $n := n + 1$.

Step 3: *If in the last* N *steps no improvement has been found, i.e., if* $C(g^{(n-N,0)}) \leq C(g^{(n-l,0)})$ *for* $l = 0, \ldots N - 1$, *then terminate.*
The best cycle found is $g^{(n-N,0)}$. *Otherwise, return to Step 2.*

5 Conclusion

A fixed cycle policy is analysed for a multi-item production system. The structure of this policy allows for a decomposition of the system into N independent periodic subsystems, one for each product type. Then an analysis is performed per product type and the optimal base stock level is found for a given fixed cycle. The analysis is extended to allow for slot dependent base stock levels. The optimal base stock levels are obtained from newsvendor type expressions.

A local search algorithm is presented that produces a (close to) optimal fixed cycle policy. This strategy will later be used to construct a good dynamic strategy by means of a single policy improvement step. Because of the decomposition of the system in independent subsystems the size of the problem in terms of product types only plays a minor role.

A Expectation in the first slot

We obtain the mean value of $\tilde{X}_1$ by taking the first derivative of the generating function $\tilde{\mathcal{G}}_1(z)$. We rewrite $\tilde{\mathcal{G}}_1(z)$ as $\frac{\mathcal{N}(z)}{\mathcal{D}(z)}$, with

$$\mathcal{N}(z) = \sum_{m=1}^{g} \sum_{k=0}^{\delta(S_m)} \tilde{p}(k,m)(z^{k+m} - z^{k+m-1})\mathcal{A}_P^{g+1-m}(z)\mathcal{A}_V(z)$$

and

$$\mathcal{D}(z) = z^g - \mathcal{A}^*(z)$$

to keep the notation simple. Here, $\mathcal{A}^*(z) = \mathcal{A}_P^g(z)\mathcal{A}_V(z)$, the generating function of the total demand during one cycle.

$\tilde{\mathcal{G}}_1'(1)$ can now be rewritten as

$$\frac{\mathcal{N}'(z)\mathcal{D}(z) - \mathcal{D}'(z)\mathcal{N}(z)|_{z=1}}{\mathcal{D}^2(z)|_{z=1}} = \frac{\mathcal{N}'(z) - \mathcal{D}'(z)\tilde{\mathcal{G}}_1(z)|_{z=1}}{\mathcal{D}(z)|_{z=1}}.$$

Since $\tilde{\mathcal{G}}_1(1) = \frac{\mathcal{N}'(1)}{\mathcal{D}'(1)}$ by l'Hôpital and $\mathcal{D}(1) = 0$, we can use l'Hôpital again:

$$\tilde{\mathcal{G}}_1'(1) = \frac{\mathcal{N}''(z) - \mathcal{D}''(z)\tilde{\mathcal{G}}_1(z) - \mathcal{D}'(z)\tilde{\mathcal{G}}_1'(z)|_{z=1}}{\mathcal{D}'(z)|_{z=1}}$$

Using $\tilde{\mathcal{G}}_1(1) = 1$ and rearranging terms gives us:

$$\tilde{\mathcal{G}}_1'(1) = \frac{\mathcal{N}''(1) - \mathcal{D}''(1)}{2\mathcal{D}'(1)},$$

with

$$\begin{aligned}
\mathcal{D}'(1) &= g - (\lambda(gT_P + T_V)\,, \\
\mathcal{N}''(1) &= 2\sum_{m=1}^{g} \sum_{k=0}^{\delta(S_m)} \tilde{p}(k,m)\left[(gT_P + T_V)\,\lambda + k + m - 1\right], \\
\mathcal{D}''(1) &= \left[g(g-1) - \mathcal{A}^{*''}(1)\right].
\end{aligned}$$

B Proof of Lemma 2

If $\rho = \sum_{i=1}^{N} \lambda_i T_{P,i} < 1$, there exists a unique stable fixed cycle of minimum length.

Proof: The proof is twofold. First it is shown – with an example – that a stable fixed cycle *exists*. Then the uniqueness of the shortest stable fixed cycle is shown by contradiction.
In order to construct a stable fixed cycle, consider the following system of linear equations:

$$g_i = \lambda_i T_{FC},\ i = 1, \ldots, N, \tag{18}$$

$$T_{FC} = \sum_{i=1}^{N} g_i T_{P,i} + \sigma. \tag{19}$$

The solution of this system is unique and (by substitution of the first equation into the second) easily seen to be $T_{FC} = \frac{\sigma}{1-\rho}$, $g_i = \lambda_i \frac{\sigma}{1-\rho}$, $i = 1 \ldots N$. Now let K satisfy

$$(K-1)\sigma > \sum_{j=1}^{N} T_j$$

and let $x = \{x_1, \ldots, x_N\}$ denote the solution of Equations (18) and (19). Define $\hat{g}(K)$ as $\{\lceil Kx_1 \rceil, \ldots\ldots, \lceil Kx_N \rceil\}$ and $\hat{T}_{FC}(K) = \sum_{i=1}^{N} \lceil Kx_i \rceil T_{P,i} + \sigma$. Then we have that

$$\hat{g}_i(K) - \lambda_i \hat{T}_{FC}(K) = \lceil Kx_i \rceil - \lambda_i \sum_{j=1}^{N} \lceil Kx_j \rceil T_j - \lambda_i \sigma$$

$$\geq Kx_i - \lambda_i \sum_{j=1}^{N} (Kx_j + 1) T_j - \lambda_i \sigma = (K-1)\lambda_i \sigma - \lambda_i \sum_{j=1}^{N} T_j > 0, \; i = 1, \ldots, N.$$

Hence $\hat{g}(K)$ describes a stable fixed cycle.

In order to prove the uniqueness, assume that there are two different stable fixed cycles of minimum length, described by $g^{(1)}$ and $g^{(2)}$, and as both represent minimal cycles, the lengths of these cycles, say $T_{FC}^{(1)}$ and $T_{FC}^{(2)}$, are equal. Now construct a new cycle by taking the minimum of the two: $g_i^{(1-2)} = \min\{g_i^{(1)}, g_i^{(2)}\}$, for $i = 1 \ldots N$. Since the cycles were different, the new cycle must be shorter, i.e., $T_{FC}^{(1-2)} < T_{FC}^{(1)} = T_{FC}^{(1)}$. But for all i the number of production slots $g_i^{(1-2)}$ for type i is already sufficient in a longer cycle (with duration $T_{FC}^{(1)}$ or $T_{FC}^{(2)}$), so the cycle $g_i^{(1-2)}$ is stable as well. But this is a contradiction, because it was assumed that $g^{(1)}$ and $g^{(2)}$ describe a stable fixed cycle of minimum length. Therefore, the shortest stable fixed cycle is unique.

C Proof of Lemma 3

Algorithm 1 produces the shortest stable fixed cycle for a polling system with N queues.

Proof: The proof is based on induction.

Assume that for $n \geq 0, g^{(n)} \leq g^{min}$, thus $T_{FC}^{(n)} \leq T_{FC}^{min}$. Then we have for all i that

$$g_i^{(n+1)} = \lfloor \lambda_i T_{FC}^{(n)} + 1 \rfloor \leq \lfloor \lambda_i T_{FC}^{min} + 1 \rfloor = g_i^{min} \; ,$$

so $g^{(n+1)} \leq g^{min}$.
As long as $g^{(n)}$ is not stable $g^{(n+1)}$ will have at least one production slot more than $g^{(n)}$ and then $T_{FC}^{n+1} > T_{FC}^{n}$. However, if $g^{(n)}$ is stable then one must have $g^{(n)} = g^{(n+1)} = g^{min}$.

It now remains to prove that $g^{(0)} \leq g^{min}$.
If one chooses $g_i^{(0)} = 1$ for all i, then obviously (assuming $\lambda_i > 0$) one has $g_i^{(0)} \leq g_i^{min}$, since for any stable cycle one has $g_i \geq 1$. This completes the proof: starting with $g^{(0)} = (1, \ldots, 1)$ the cycles $g^{(n)}$ monotonically increase until the minimal cycle has been found.

One may speed up the algorithm a little by choosing a different vector for $g^{(0)}$. In order to see this, assume that T is the duration of some stable fixed cycle. Then the corresponding g_i must satisfy $g_i > \lambda_i T$ for all i. Thus $\sum_i g_i T_{P,i} > \sum_i \lambda_i T T_{P,i}$, or $T - \sigma > \rho T$, so $T > \frac{\sigma}{1-\rho}$. Now define $g_i^{(0*)} = \left\lfloor \lambda_i \frac{\sigma}{1-\rho} \right\rfloor$ for all i. One easily sees that $g_i^{(0*)} \leq g_i^{min}$ (since $T_{FC}^{(0*)} < T_{FC}^{min}$), and $g_i^{(1)} > g_i^{(0*)}$ for the $g^{(1)}$ constructed from $g^{(0*)}$ along the lines of the algorithm.

References

[1] Adan, I.J.B.F., J.S.H. van Leeuwaarden and E.M.M. Winands (2006). On the application of Rouché's theorem in queueing theory. *Operations Research Letters.* **34**: 355–360.

[2] S. Bhulai (2004). Dynamic routing policies for multi-skill call centers. *Technical Report WS2004-11.* Vrije Universiteit, Amsterdam.

[3] Broek, M.S. van den, J.S.H van Leeuwaarden, I.J.B.F. Adan and O.J. Boxma (2006). Bounds and approximations for the fixed-cycle traffic light queue. *Transportation Science.* **40(4)**: 484–496.

[4] Bruin, J. and J. van der Wal. A MDP decomposition approach on multi-item production systems (in preparation).

[5] Bruneel, H. and B.G. Kim (1993). *Discrete-Time Models for Communication Systems including ATM*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

[6] C Buyukkoc, P Varaiya and J Walrand (1985). *The $c\mu$-rule revisited* Advances in Applied Probability, **17(1)**: 237–238.

[7] Darroch, J.N. (1964). On the traffic light queue, *Annals of Mathematical Statistics* **35**: 380–388.

[8] Denteneer, D., J.S.H. van Leeuwaarden and I.J.B.F. Adan (2007). The acquisition queue, *Queueing Systems* **56**: 229–240.

[9] van Eenige, M.J.A. (1996). *Queueing Systems with Periodic Service*, Ph.D. thesis, Technical University Eindhoven, Eindhoven.

[10] Erkip, N., R. Güllü and A. Kocabiyikoglu (2000). A quasi-birth-and-death model to evaluate fixed cycle time policies for stochastic multi-item production/inventory problem, *Proceedings of MSOM conference, Ann Harbor.*

[11] Federgruen, A. and Katalan, Z. (1996). The stochastic economic lot scheduling problem: cyclical base-stock policies with idle times, *Management Science* **42 (6)**: 783–796

[12] Fuhrmann, S.W. and R.B. Cooper (1985). Stochastic decomposition in an M/G/l queue with generalized vacations, *Operatoins Research* **33**: 11–17.

[13] Güllü, R., N. Erkip and S. Hafizogullari. Fixed Cycle Time Policies for the Stochastic Multi-Item Production/Inventory Problem.

[14] Haijema, R. and J. van der Wal (2008). An MDP decomposition approach for traffic control at isolated signalized intersections, *Probability in the Engineering and Informational Sciences* **22(4)**: 587–602.

[15] Howard, R. (1960). Dynamic Programming and Markov Processes. *The MIT Press*, Cambridge, Mass.

[16] Janssen, A.J.E.M., J.S.H. van Leeuwaarden (2004). Analytic computation schemes for the discrete-time bulk service queue, *Queueing Systems* **50**: 141–163.

[17] van Leeuwaarden, J.S.H. (2006). Delay Analysis for the Fixed-Cycle Traffic-Light Queue, *Transportation Science* **40**: 189–199.

[18] P. van Mieghem (1996). The asymptotic behavior of queueing systems: Large deviations theory and dominant pole approximation *Queueing Systems*, **23**: 27–55.

[19] Norman, J.M. (1972). Heuristic procedures in dynamic programming, *Manchester University Press, Manchester*

[20] Pakes, A.G. (1969). Some conditions for ergodicity and recurrence of Markov chains,*Operations Research* **17**: 1058–1061.

[21] Porteus, E.L. (2002). Foundations of stochastic inventory, *Stanford University Press, Stanford*

[22] Resing, J.A.C.(1993). Polling systems and multitype branching processes,*Queueing Systems* **30**: 409–426.

[23] Sassen, S.A.E., H. C. Tijms and R. D. Nobel (1997). A heuristic rule for routing customers to parallel servers *Statistica Neerlandica* **51 (1)**: 107–121.

[24] Takagi, H. (1988). Queueing Analysis of Polling Models,*ACM Computing Surveys* **20**: 5–28.

[25] Tijms, H.C. and M.C.T. van de Coevering (1991). A simple numerical approach for infinite-state Markov chains, *Probability in the Engineering and Informational Sciences* **5**: 85–295.

[26] Wijngaard, J. (1979). Decomposition for dynamic programming in production and inventory control, *Engineering and Process Economics* **4**: 385–388.

[27] Winands, E.M.M., I.J.B.F. Adan and G.J. van Houtum (2005). The Stochastic Economic Lot Scheduling Problem: A Survey, *BETA WP* Eindhoven: BETA WP-133, Beta Research School for Operations Management and Logistics; 2005.