Available online at www.sciencedirect.com
ScienceDirect
Discrete Applied Mathematics 156 (2008) 412–419
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

A polynomial time algorithm for solving a quality control station configuration problem

Michal Penn, Tal Raviv

Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology, Haifa 31905, Israel

Received 2 August 2007; accepted 2 August 2007
Available online 21 September 2007

Abstract

We study unreliable serial production lines with known failure probabilities for each operation. Such a production line consists of a series of stations; existing machines and optional quality control stations (QCS). Our aim is to simultaneously decide where and if to install the QCSs along the line and to determine the production rate, so as to maximize the steady state expected net profit per time unit from the system.

We use dynamic programming to solve the cost minimization auxiliary problem where the aim is to minimize the time unit production cost for a given production rate. Using the above developed $O(N^2)$ dynamic programming algorithm as a subroutine, where N stands for the number of machines in the line, we present an $O(N^4)$ algorithm to solve the Profit Maximization QCS Configuration Problem.

Keywords: Quality control; Serial production line; Polynomial algorithm

1. Introduction

In this paper we consider the well-studied problem of incorporating quality control stations (QCSs) (some times referred to as inspection stations) into unreliable multi stage production systems. The effect of insertion of QCSs on the final cost and on the quality of the final product is well observed in the literature. Models and optimization algorithms for various problems related to locating QCS stations along production lines are date back to 1965, see Lindsay and Bishop [2]. A survey on the problem of optimal location of QCS along multi stage systems appears in Raz [4]. For more recent studies focused on systems with imperfect inspection facilities, see for example Raz and Kaspi [5], for a study on systems allowing rework and repair see Yum and McDowell [7]. All the above studies consider optimization of steady state performance. For the problem under a finite planning horizon setting look at Kogan and Raz [1].

We point out that the above-mentioned studies focus on maximizing the profit per product and overlook the effect of the QCS configuration on the system throughput and on the holding costs of work in process. In [3] we consider holding costs and utilize the influence of the QCS configuration on the production line throughput in the branch and bound strategy developed there, one that maximizes the expected profit per time unit. To the best of our knowledge, this was the first attempt to maximize the expected profit per time unit rather than per finished product. We continue to follow this line in the present paper.

E-mail addresses: mpenn@ie.technion.ac.il (M. Penn), talraviv69@gmail.com, talraviv@eng.tau.ac.il (T. Raviv).

0166-218X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2007.08.005

In this paper we look for an optimal QCS configuration in a serial production line, in a steady state, under any arrival process and with zero holding costs. Two optimization problems are considered: Minimization of the expected operational cost under a given production rate and maximization of the expected profit per time unit where the QCS configuration and the production rate are to be decided simultaneously. As opposed to our approximation branch and bound strategy in [3], where holding costs were taken into account, here we provide an exact polynomial time algorithm. We first solve the cost minimization problem using an $O(N^2)$ time dynamic programming algorithm, where N stands for the number of machines along the line. Then, the profit maximization problem is solved by an $O(N^4)$ time algorithm that uses the cost minimization algorithm as a subroutine. A key ingredient in our proof of the complexity of the maximization algorithm is the $O(N^2)$ bound on the number of potential maximum eligible production rates, and an efficient way to obtain these rates.

The rest of the paper is organized as follows. In Section 2 we define the two versions of our QCS problem. Section 3 presents a dynamic programming algorithm to solve the cost minimization problem and Section 4 provides a polynomial time algorithm for the profit maximization problem. We conclude with a short discussion in Section 5.

2. Preliminaries and notations

Consider a serial production line with N machines and an infinite number of identical products to be produced. For simplicity of exposition we slightly abuse the notion of a product and, if no confusion may arise, we refer to an unfinished product as a product. Processing of a product consists of a series of N operations, with the ith operation processed on the ith machine. The ith operation's cost is c_i and its processing time assumed to be an i.i.d random variable with expectation x_i. Operation i, if performed on a *non-defective* product, succeeds with a known probability p_i and fails with probability $(1 - p_i)$. A defective product processed by any machine remains defective.

QCSs can be installed anywhere along the line and each installed QCS detects all the defective products delivered from its preceding machine and discards them from the line. Machines and QCSs are referred to as stations. Each product can be processed by a single station at a time, and each station can process one product at time. An unlimited buffer is located in front of each station where all products that finished their previous operations are waiting to be processed. A fictitious buffer in front of the first machine represents the products that entered the production line but their first operation has not yet been started. The arrival process to the fictitious buffer is assumed to be any stationary stochastic process, including the deterministic process with constant inter-arrival times. The rate of the arrival process into the fictitious buffer is referred to as the *production rate* and is denoted by a. Note, however, that in general, installed QCSs reduce the output rate to be less than a since part of the products become defective and are thus discarded from the system.

The ith machine is denoted by M_i and its immediate consecutive QCS, if such installed, is denoted QC_i. The location of the last installed QCS before M_i is denoted by L_i, with the convention that $L_i = 0$ if no such QCS is installed, and L_{N+1} is the location of the last QCS in the line. Any QCS configuration is denoted by a set Y of its installed QCSs. For convenience, when it is clear from the context, we refer to Y as the characteristic vector of this set. That is, $Y_i = 1$ if QC_i exists and $Y_i = 0$ otherwise. Note that L_i is determined by Y and thus the notation $L_i(Y)$ should be used. However, to avoid cumbersome notation, we use L_i rather than $L_i(Y)$, if the particular configuration Y is clear from the context. We use B_i (B_i') to denote the ith corresponding buffer in front of M_i (QC_i). Note that B_1 is the fictitious buffer. The cost of an inspection done by QC_i, if such installed, is c_i' and its length assumed to be an i.i.d random variable with mean x_i'. In addition, there is a time unit fixed capital cost of f_i' associated with each installed QCS, regardless of its actual working rate. Capital costs of machines are considered as sunk costs and thus can be eliminated form the optimization process hereinafter. Each non-defective (defective) finished product has its own revenue (penalty cost) denoted by r_G (r_B).

Let q_{ij} denote the probability that a non-defective product leaving machine M_i remains non-defective while leaving M_j. Clearly,

$$q_{ij} \equiv \prod_{k=i+1}^{j} p_k. \tag{1}$$

Note that (1) holds also if the failure events are dependent across machines since the p_i''s are the probabilities that a product will come out of M_i as non-defective, conditioned on being non-defective when it entered M_i. Note that q_{0i} stands for the probability that a product was successfully processed along the partial series of machines $M_1, \ldots, M_i$.

Consider a QCS system with a given QCS configuration Y. Clearly, if the steady state arrival rate into B_i is a_i, and there is no installed QCS between the two consecutive machines M_i and M_{i+1}, then the arrival rate into B_{i+1} is

$$a_{i+1} = \min\left\{a_i, \frac{1}{x_i}\right\}.$$

Furthermore, if QC_i exists, then

$$a_{i+1} = q_{L_i,i} \cdot \min\left\{a_i, \frac{1}{x_i}, \frac{1}{x'_i}\right\}.$$

Recall that $q_{L_i,i}$ indicates the probability that a product remains non-defective after completed its ith operation, given L_i, the location of the last installed QCS before M_i. Hence, $q_{L_i,i}$ stands for the proportion of products qualified by QC_i.

A production rate a is said to be *eligible* for a given QCS configuration Y, if its implied arrival rate to each station along the line is at most the station's potential production (inspection) rate. That is, $a_i \leqslant \frac{1}{x_i}$ and $Y_i \cdot a_i \leqslant \frac{1}{x'_i}$ for all $i = 1, \ldots, N$. Observe that it is undesirable to operate a system under any production rate that exceeds the maximum eligible rate. This is since the throughput of non-defective products associated with the maximum eligible rate dominates any other such throughput from the system. A QCS configuration Y is said to be *eligible* for a production rate a, if a is eligible for Y. Using these notations, if a is eligible for Y, then the throughput rate of the products (non-defective products) is $aq_{0,L_{N+1}}$ $(aq_{0,N})$.

Given a QCS configuration Y and a production rate a, we denote by $C(Y, a)$ the time unit steady state operational cost of the system. The convention $C(Y, a) = \infty$ is used if a is ineligible for Y. For a given production rate a, let $Y^*(a)$ denote an optimal configuration that minimizes $C(Y, a)$ and $C^*(a) = C(Y^*(a), a)$ denotes this minimal cost.

Two problems are considered in this paper, the auxiliary cost minimization problem and the main problem of profit maximization. The problem of determining an optimal QCS configuration for a given production rate, one that minimizes the steady state expected production cost per time unit, is referred to as the *Cost Minimization QCS Configuration Problem*. The second problem of maximizing the steady state expected net profit from the system per time unit by simultaneously determining the QCS configuration and the production rate is referred to as the *Profit Maximization QCS Configuration Problem*. A solution of the Profit Maximization Problem is given by (Y, a), a pair of a QCS configuration Y and an eligible production rate a for it. In the following, the tuple $(\mathbf{p}, \mathbf{x}, \mathbf{x}', \mathbf{c}, \mathbf{c}', \mathbf{f}', \mathbf{r_B}, \mathbf{r_G})$ is referred to as a *QCS system*.

3. The cost minimization problem

In this section we present our quadratic dynamic programming algorithm for solving the Cost Minimization QCS Configuration Problem that returns an optimal QCS configuration for a given production rate.

Algorithm 3.1. (*Cost Minimization Algorithm*).

Input: *A QCS system* $(\mathbf{p}, \mathbf{x}, \mathbf{x}', \mathbf{c}, \mathbf{c}', \mathbf{f}', \mathbf{r_B}, \mathbf{a})$.[1]

Output: $Y^*(a)$, $C^*(a)$.

We use L_i as state variable and Y_i as a decision binary variable that indicates whether or not to install QC_i. The recursion formula $q_{ij} = q_{i,j-1} p_j$ is used to calculate the q_{ij}'s. The function $h_i(L_i; Y_i)$ returns the cost incurred by the tail of the line for a given (L_i, Y_i). It is recursively constructed as follows:

$$h_i(L_i; Y_i) = \begin{cases} aq_{0,L_i}(c_i + c'_i Y_i) + f'_i Y_i + h^*_{i+1}(L_{i+1}(L_i, Y_i)) & aq_{0,L_i} \in \left[0, \min\left(\frac{1}{x_i}, \frac{1-Y_i}{x_i} + \frac{1}{x'_i}\right)\right], \\ \infty & otherwise. \end{cases} \quad (2)$$

Note that the upper bound on the flow rate in each step is set to $\frac{1}{x_i} = \min(\frac{1}{x_i}, \frac{1-Y_i}{x_i} + \frac{1}{x'_i})$ *for* $Y_i = 0$, *and to* $\min(\frac{1}{x_i}, \frac{1}{x'_i})$ *for* $Y_i = 1$. *We use the following transition function*:

$$L_{i+1}(L_i, Y_i) = \begin{cases} L_i, & Y_i = 0, \\ i, & Y_i = 1. \end{cases} \quad (3)$$

[1] Note that r_G, the revenue per product, is irrelevant for this problem.

The initial condition for h_N *is*

$$h_N(L_N;Y_N) = aq_{0,L_N}[(1-Y_N)\cdot r_B\cdot(1-q_{L_N,N}) + (c_N + c'_N\cdot Y_N)] + Y_N f'_N \tag{4}$$

if $aq_{0,L_N} \in [0, \min(\frac{1}{x_N}, \frac{1-Y_N}{x_N} + \frac{1}{x'_N})]$, *and* $h_N(L_N;Y_N) = \infty$ *otherwise.*

The function h_i^* *is constructed by*

$$h_i^*(L_i) = \min_{Y_i} h_i(L_i;Y_i), \tag{5}$$

and the value of $h_1^*(0)$ *is returned as* $C^*(a)$. *If* $h_1^*(0)=\infty$, *then there is no eligible configuration for the given production rate* a. *In such a case*, $Y^*(a)=\emptyset$ *and* $C^*(a)=\infty$. *Otherwise*, *the optimal QCS configuration is determined in the forward iterations by*

$$Y_i^*(L_i) \in argmin_{Y_i} h_i(L_i;Y_i). \tag{6}$$

Proposition 3.2. *Algorithm* 3.1 *is correct and its time and space complexity are* $\mathrm{O}(N^2)$.

Proof. The correctness of Algorithm 3.1 follows directly from Bellman's principle of conditional optimization, see for example [6]. Calculating q using the recursion formula $q_{ij} = q_{i,j-1}p_j$, for all $i > j$, takes $\mathrm{O}(N^2)$ operations and the space required to store q is $\mathrm{O}(N^2)$. In addition, at any backward iteration i, the function $h_i(L_i;Y_i)$ is calculated in a constant number of operations for the two possible values of Y_i ($\{0,1\}$) and for i possible values of L_i ($\{0,\ldots,i-1\}$). Thus, there are $N(N+1)$ such calculations in total. The forward iterations to determine the optimal configuration, take $\mathrm{O}(N)$. Hence, the overall time complexity of the algorithm is $\mathrm{O}(N^2)$. The results of each stage i are stored in i reals $[h_i^*(L_i)]$ and i boolean variables $[Y_i^*(L_i)]$ and thus the total space complexity is $\mathrm{O}(N^2)$ as well. $\square$

4. The Profit Maximization Problem

Here we present an $\mathrm{O}(N^4)$ time algorithm for solving the Profit Maximization QCS Configuration Problem that uses Algorithm 3.1 as a subroutine. The key observation for proving the complexity of the Profit Maximization Algorithm is Proposition 4.3 below that enables us to bound nicely the number of times Algorithm 3.1 is executed in one run of the Profit Maximization Algorithm.

Observe that if (Y,a) is a solution of the Profit Maximization QCS Configuration Problem, then the expected profit from the system per time unit is given by

$$P(Y,a) = a\left\{q_{0,N}r_G - (1-Y_N)\cdot(1-q_{N,L_N})r_B - \sum_{i=1}^{N}[q_{0,L_i}(c_i + c'_iY_i)]\right\} - \sum_{i=1}^{N} Y_i f'_i \tag{7}$$

if a is an eligible production rate for the configuration Y. Otherwise, $P(Y,a)$ is undefined (sometimes we use the notation $P(Y,a) = -\infty$ to indicate this). It is apparent from the above exposition of $P(Y,a)$, in (7), that for a given QCS configuration Y, $P(Y,a)$ is a linear function of a within its domain. Its slope is given by

$$q_{0,N}r_G - (1-Y_i)\cdot(1-q_{0,L_N})r_B - \sum_{i=1}^{N}[q_{0,L_i}(c_i + c'_iY_i)]$$

and its offset is

$$-\sum_{i=1}^{N} Y_i f'_i.$$

Consider now the function

$$P^*(a) \equiv \max_Y P(Y,a) = a\cdot q_{0,N}\cdot r_G - C^*(a). \tag{8}$$

Using $P^*(a)$, our Profit Maximization QCS Configuration Problem can be stated as $\max_a P^*(a)$.

Table 1
Data for Example 4.1

Station #	p	x	x'	c	c'	f'
1	0.8	10	9	4	1	0.1
2	0.8	13	12	6	1	0.3
3	0.8	14	14	6	1	0.4
4	0.85	16	17	8	1	0.8

$r_G = 80$, $r_B = 10$.

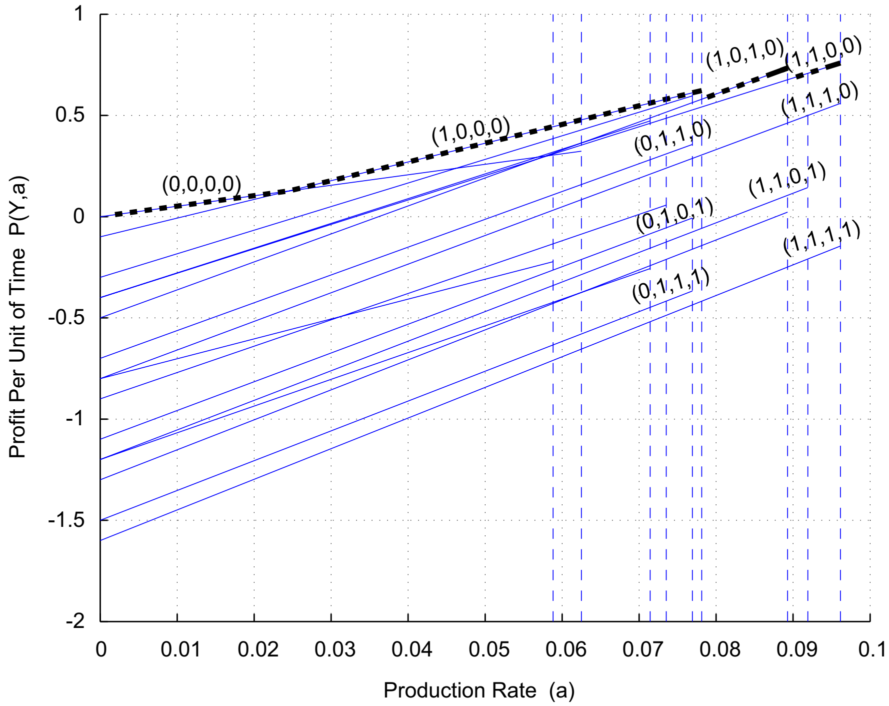

Fig. 1. The functions $P(Y, a)$ are plotted for all 16 possible configurations of the QCS problem presented in Example 4.1. The function $P^*(a)$ is plotted in thick dashed line. The vectors above some of the $P(Y, a)$ functions are their corresponding QCS configurations.

Note that $P^*(a)$ describes the maximum attainable profit from a QCS system for a given production rate a. This function is a piecewise linear (not necessarily continuous) function. This is since it is obtained as a maximization over the set of the linear functions $P(Y, a)$ for all 2^N possible QCS configurations. Example 4.1 with its Fig. 1 to follow, illustrate the structure of $P^*(a)$.

Example 4.1. Consider the following QCS system that consists of four machines, four optional QCS and the data presented in Table 1. Fig. 1 illustrates the 16 possible $P(Y, a)$ functions, one for each QCS configuration, and the obtained $P^*(a)$ function.

Observation 4.2. *Let (Y^*, a^*) be an optimal solution of the Profit Maximization QCS Configuration Problem that induces a positive profit. Then, this solution satisfies,*

1. *$P(Y^*, a)$ is an increasing function of a in the relevant domain. That is,*

$$q_{0,N} r_G - (1 - Y_N^*(a)) \cdot (1 - q_{0,L_N}) r_B - \sum_{i=1}^{N} [q_{0,L_i}(c_i + c_i' Y_i^*(a))] > 0.$$

2. *Let $U(Y)$ denote the maximum eligible production rate allowed by the QCS configuration Y, then $a^* = U(Y^*)$.*

Observe that if none of the (Y, a)'s can achieve positive profit, then $P(Y^*, a)$ is a non-increasing function of a and hence, the optimal solution is not to produce at all (that is $a^* = 0$). For a given system, let $\mathscr{U} = \bigcup_{Y \in \{0,1\}^N} \{U(Y)\}$ be the set of all maximum eligible production rates for all possible QCS configurations. Then, Observation 4.2 implies that optimizing $P^*(a)$ over $\mathscr{U}$ is sufficient for solving the Profit Maximization QCS Configuration Problem. Furthermore, although the number of possible QCS configurations is 2^N, Proposition 4.3 below shows that $|\mathscr{U}|$ is $O(N^2)$. This crucial observation is the key for proving the polynomiality of our Profit Maximization Algorithm.

Proposition 4.3. *For any given QCS system, the number of maximum eligible production rates for all possible configurations is at most* $\binom{N+2}{2}$. *That is,* $|\mathscr{U}| \leqslant \binom{N+2}{2}$.

Proof. First note that if $a = U(Y)$, then there must be at least one station (either a machine or a QCS) that under (Y, a) works in its full capacity. We call such station a *bottleneck*. Given a QCS system and a QCS configuration Y, the bottleneck station with the smallest index is said to be the *first bottleneck* of the production line under $(Y, U(Y))$, and is denoted by $S(Y)$. We use the notation S for $S(Y)$ if the configuration Y is clear from the context.

For each pair $0 \leqslant i < j \leqslant n + 1$, the set of configurations $\mathscr{Y}_{ij}$ includes all the configurations Y with S being one of $M_{i+1}, \ldots, M_j, QC_j, L_j = i$ and $Y_j = 1$ for $j < N + 1$.

Clearly, the set of all 2^N possible QCS configurations can be partitioned according to these $\binom{N+2}{2}$ subsets. Now, our proof follows from the fact that for any pair (i, j), all its associated $U(Y)$'s are of equal value for all QCS configurations $Y \in \mathscr{Y}_{ij}$. The above fact is argued as follows. First note that the pair (i, j) uniquely determines S if $\mathscr{Y}_{ij} \neq \emptyset$. This is since S is a slowest station with the smallest index among $M_{i+1}, \ldots, M_j, QC_j$. Once S is identified, the maximum eligible production rate is uniquely determined. Let $\bar{x}$ denote the expected processing time of S. Then, $(\bar{x} \cdot q_{ij})^{-1}$ is the maximum eligible production rate for each configuration $Y \in \mathscr{Y}_{ij}$. To see this, observe that the arrival rate to each of the stations $M_{i+1}, \ldots, M_j, QC_j$ is $aq_{0,i}$ which is equal to $\frac{1}{\bar{x}}$, the production rate of S. We conclude that each subset $\mathscr{Y}_{ij}$ contributes at most one member to $\mathscr{U}$ and so $|\mathscr{U}| \leqslant \binom{N+2}{2}$. $\square$

The basic idea behind the above proof is illustrated by Example 4.1 and Fig. 1. Consider for example the three QCS configurations (1,1,0,0), (1,1,1,0), and (1,1,1,1). Observe that their corresponding three $P(Y, a)$ profit functions terminate at the same value of a, as demonstrated by the vertical dashed lines in Fig. 1, and their corresponding first bottleneck station is M_2. These configurations form the set $\mathscr{Y}_{1,2}$. Similarly, $\mathscr{Y}_{0,2} = \{(0, 1, 0, 1), (0, 1, 1, 1), (0, 1, 1, 0)\}$. Here the first bottleneck station is QC_2 and no QCS precedes it. Recall that the number of possible configurations grows exponentially with the number of stations and thus, clearly, it is impractical to maximize $P^*(a)$ directly over all possible $P(Y, a)$'s. We are now ready to describe our main algorithm.

Algorithm 4.4. (*Profit Maximization Algorithm*).
Input: *A QCS system* $(\mathbf{p}, \mathbf{x}, \mathbf{x}', \mathbf{c}, \mathbf{c}', \mathbf{f}', \mathbf{r_B}, \mathbf{r_G})$.

Output: (Y^*, a^*) *a pair of a QCS configuration and an eligible production rate that maximizes* $P(Y, a)$.
`Let` $Y^* = \emptyset$ `and` $a^* = 0$;
`For each pair of integers` (i, j) `with` $0 \leqslant i < j \leqslant N + 1$
`Let` $\bar{x}$ `be the expected processing time of the first slowest station among` $M_{i+1}, \ldots, M_j, QC_j$
(Or $M_{i+1}, \ldots, M_N$ `for the case` $j = N + 1$);
`Calculate the corresponding maximum production rate` $a = \frac{1}{\bar{x} \cdot q_{0,i}}$;
`Call Algorithm 3.1 to obtain` $Y^*(a)$ `and` $C^*(a)$;
`Calculate` $P^*(a)$ `as in (8)`;
`If` $P^*(a) > P(Y^*, a^*)$ `Then`
`Let` $a^* = a$ `and` $Y^* = Y^*(a)$;
`Return` (Y^*, a^*);

Theorem 4.5. *Algorithm* 4.4 *is correct with time complexity of* $O(N^4)$ *and space complexity of* $O(N^2)$.

Proof. For each configuration set $\mathcal{Y}_{ij}$ we calculate the maximum eligible production rate a and then call Algorithm 3.1 to obtain $Y^*(a)$ and $P^*(a)$. We observe that $P^*(a)$ is at least as good as the value of the best solution attained by any configuration in $\mathcal{Y}_{ij}$. The correctness of the algorithm follows from the above coupled with the fact that $\{\mathcal{Y}\}$ is a partition of all possible QCS configurations.

Now, by Proposition 4.3, Algorithm 4.4 calls Algorithm 3.1 at most $O(N^2)$ times. Recall that by Proposition 3.2, the complexity of Algorithm 3.1 is $O(N^2)$ and hence we obtain the overall time complexity of $O(N^4)$. The space complexity of Algorithm 3.1 is $O(N^2)$ and since the same memory can be reused at each call of this algorithm, the space complexity of Algorithm 4.4 is $O(N^2)$ as well. □

A closer look at some of the properties of the QCS systems enables us to further reduce the processing time of Algorithm 4.4. However, as for now, those reductions do not improve the computational complexity of the algorithm. We demonstrate such a possible improvement by the following simple observations presented in Proposition 4.6 below. These observations enable us to exclude from further computation some of the (i, j) pairs, pairs that correspond to empty subsets $\mathcal{Y}_{ij}$, and hence to reduce the actual running time of Algorithm 4.4.

Proposition 4.6. *Consider a pair (i, j) and let $\bar{x}$ denote the expected processing time of $\bar{S}$, the slowest station among $M_{i+1}, \ldots, M_j, QC_j$. If one the following conditions holds, then $\bar{S}$ is not the first bottleneck along the production line, and thus $\mathcal{Y}_{ij} = \emptyset$.*

1. $\bar{x} \leqslant \frac{x_k}{q_{ki}}$ *for some* $k \leqslant i$,[3pt]
2. $\bar{x} \leqslant \frac{x_i'}{p_i}$, *and*[3pt]
3. $\bar{x} > q_{ik} x_k$ *for some* $k > j$.

Proof. Assume by contradiction that a configuration $Y \in \mathcal{Y}_{ij}$ for which the first condition holds exists. That is, the first bottleneck is located between QC_i and QC_j (including QC_j but not including QC_i) and for some $k \leqslant i$ we have that

$$\frac{1}{\bar{x}} \geqslant \frac{q_{ki}}{x_k}.$$

Note that, for configuration Y, the maximum eligible flow rate via the stations $M_{i+1}, \ldots, M_j, QC_j$ is $U(Y) \cdot q_{0i}$. Thus,

$$U(Y) \cdot q_{0i} = \frac{1}{\bar{x}}$$

and so

$$U(Y) \cdot q_{0i} \geqslant \frac{q_{ki}}{x_k}. \tag{9}$$

Dividing both sides of (9) by q_{ki} and using the definition of q as in (1) we get

$$U(Y) \cdot q_{0k} \geqslant \frac{1}{x_k}.$$

However, this implies that using the maximum eligible flow rate for configuration Y causes the flow rate via M_k to be at least its capacity $\frac{1}{x_k}$; contradicting the fact that the first bottleneck of the line is located between QC_i and QC_j. This completes the correctness of the first condition.

Similarly, assume by contradiction that a configuration $Y \in \mathcal{Y}_{ij}$ for which the second condition holds exists. That is,

$$\frac{1}{\bar{x}} \geqslant \frac{p_i}{x_i'}.$$

As before we have $\frac{1}{\bar{x}} = U(Y) q_{0i}$ and so

$$U(Y) q_{0i} \geqslant \frac{p_i}{x_i'}.$$

By the definition of q we have that $q_{0i} \geqslant p_i$ and so $U(Y) \geqslant \frac{1}{x_i'}$ which is again a contradiction to the fact that the first bottleneck is located after QC_i.

The proof of the third case is very similar to the first one and thus omitted. It should be noted that the strict inequality in the third case follows from the fact that M_k may work in its full capacity and still not be the **first** bottleneck. □

5. Discussion

We present in this paper a method to optimize, in a steady state, an unreliable serial production line by considering the possibility of installing QCSs along the line. Our results hold for any arrival process and under the assumption that no holding costs incurred by work in process. If holding costs are relatively high, we suggest the use of the approximation branch and bound method presented in [3].

We first present a simple $O(N^2)$ dynamic programming algorithm that minimizes the expected cost per time unit under a specified production rate. We then show how to use this algorithm in order to obtain simultaneously a pair of an optimal QCS configuration and its appropriate production rate, so as to maximize the expected profit per time unit. The basic idea behind our $O(N^4)$ maximization algorithm is the observation that the size of the set of possible values for optimal production rates is relatively small and an efficient method to identify this set.

Our solutions imply full utilization of the bottleneck stations. Thus, implementing our solutions "as is" results in an unstable system since the arrival rates to the bottleneck stations equal their production rates. Nevertheless, using the following minor modifications one can "stabilize" the obtained solution. In the cost minimization problem one should solve the problem for a slightly higher production rate than the one required. For the profit maximization problem, the actual production rate to be used for the optimal QCS configuration obtained should be slightly lower than the one obtained by the algorithm.

For further research we point out the following various possible extensions and generalizations of the problems presented in this paper. To extend our model to allow unreliable QCSs, rework, scrap values, etc. Further on, we suggest introducing QCS to more general production models such as job shop, multistage shops, assembly lines, etc. Another possible direction is to remove our independence assumption, that of the failure events. This calls for more sophisticated quality control methods such as sample inspection. It will also be interesting to explore an adaptive QCS policy where the actual inspection done by an installed QCS should be decided on-line by considering the current state of the system.

References

[1] K. Kogan, T. Raz, Optimal allocation of inspection effort over a finite planning horizon, IIE Trans. 34 (2002) 515–527.
[2] G.F. Lindsay, A.B. Bishop, Allocation of screening inspection effort: a dynamic programming approach, Management Sci. 10 (1965) 342–352.
[3] M. Penn, T. Raviv, Optimizing the quality control station configuration, Nav. Res. Log. 54 (2007) 301–314.
[4] T. Raz, A survey of models for allocating inspections effort in multistage production system, J. Quality Technol. 18 (1986) 239–247.
[5] T. Raz, M. Kaspi, Location and sequencing of imperfect operations in serial multi-stage production system, Internat. J. Production Res. 29 (1991) 1654–1659.
[6] M. Sniedovich, Dynamic Programming, Marcel Dekker, New York, 1992.
[7] B.J. Yum, E.D. McDowell, Optimal inspection policies in a serial production system including scarp, rework and repair: an MILP approach, Internat. J. Production Res. 25 (1987) 1451–1464.