PERFORMANCE MEASURES OF A MULTIPLE-PRODUCT PRODUCTION LINE

The production line under study is composed of a sequence of workstations connected in series with finite buffers in between. To reduce blocking and starvation, a nonlinear mathematical programming model is designed to achieve the best performance measures subject to available station capacity.

Introduction

Economic pressures and the global market forces demand increasing throughput without much changes in the capacity of production machinery. We shall consider an automated production line composed of workstations connected in series with finite buffers in between. The machines at these workstations are assumed to produce at a known constant rate. The random failures of workstations are not explicitly modeled. In order to compensate for downtimes the processing times of products are inflated by a factor $A_i$, $0 < A_i \leq 1$, where $A_i$ is the long term availability of the workstation $i$. The inflated processing time equals the true processing time divided by $A_i$. Due to the multi-product nature of the production, and existence of variation in processing times from one workstation to another, buffers are used to lower the impact of such variation by reducing the frequency of blocking and starvation of the workstations. Blocking occurs when the buffer after a workstation is full and the subsequent workstation does not drain it fast enough to make room for more products to come in. Starvation occurs when the buffer before a workstation is empty and the upstream workstation does not feed it fast enough to keep the downstream workstation in operation. The production work is done in batches of different product types. There is a high production volume of each product type but the variety of product types is limited to a handful. The production follows the same sequence of workstations for all product types. A certain amount of setup time is required for each workstation as the production is switched from one product type to another. Each workstation consists of one machine that will process one product type at a time, which then becomes the input to the next workstation via the intermediate buffer. For the sake of simplicity, the terms machine, station and workstation are used interchangeably throughout this paper.

Figure 1 depicts the production line under study where there are $m$ workstations buffered with $(m-1)$ buffers. The line processes $n$ different product types in a predetermined sequence. In
this paper, the terms product type and batch are used interchangeably. The flow direction is indicated by the arrows.

**Figure 1**

The Production Line Under Study

The aim of this paper is to develop a mathematical program to decide the best buffer size for optimal throughput and provide the production manager to perform necessary sensitivity analysis. The model is nonlinear and integer in nature.

**Literature Review**

We are interested in examining the impact of buffer size on the throughput and the makespan. The published work does not examine the problem from our perspective. Mostly, they deal with optimization of output for a given buffer arrangement.

Roser et al. (2003) studied the effect of buffers on the throughput using simulation. Li and Huang (2005) utilize a system-theoretic approach referred to as “overlapping decomposition” to model and analyze the performance of multiple-product manufacturing systems. It was proved, analytically, that the iterations would converge. The accuracy of the model’s output was estimated using a numerical method. Van Nyen et al. (2005) proposed a three-step heuristic to coordinate production and inventory control decisions in an integrated multi-product multi-machine production-inventory system characterized by job shop routings and stochastic demand, set-up and processing times. This heuristic minimizes the sum of set-up costs, work-in-process holding costs and final inventory holding costs while stochastic customer demand is satisfied with a target fill rate.

Buffer allocation is another area of the published research. Gershwin and Schor (2000) provide an efficient algorithm for determining how buffer space should be allocated. Spinellis and Papadopoulos (2000) described a simulated annealing approach for solving the buffer allocation problem in reliable production lines. The objective of this model is maximization of the average throughput through determination of near optimal buffer allocation. Shi and Men (2003) emphasize the importance of the optimal buffer allocation for production lines. They offer a hybrid Nested Partitions (NP) and Tabu Search (TS) method to solve this complex problem. Smith and Cruz (2005) provide an approximation formula for predicting the optimal buffer
allocation. The formula is developed based on a two-moment approximation formula involving the expressions for M/M/1/K and M/G/1/K systems.

The current trends in modeling manufacturing systems by using heuristics and evolutionary algorithms have some limitations in terms of model’s development time, convergence, and quality of results. These methods are not easy to explain to the decision makers and are not well accepted in the industry. It is evident that it is advantageous to implement a more comprehensive modeling approach to easily address various measures of performance of multi-product multi-stage production line and provide answers to many questions that a decision-maker may have in terms of buffer sizes, throughput, makespan, available time for maintenance, and others measures used in manufacturing systems.

We shall present problem formulation in section 3, an application and a numerical example in section 4, and provide concluding remarks in section 5 of this paper.

The Model

The Following mathematical model is formulated for a production line that produces a handful of high volume product types. Each product \( j \) is produced in batches of size \( N_j \) and goes through each of the \( m \) workstations. A buffer capacity of \( b_i \) is utilized between workstations \( i \) and \((i + 1)\) to lower the frequency of blocking and starvation.

Items produced are transferred to the downstream workstation individually. This is in contrast to the batch transfer arrangement, where all items of a product are transferred after the entire batch is completed on a workstation.

The production throughput is positively affected by larger buffer sizes. This effect shall be examined in the coming passages. Any time saved in a station may be utilized for preventive maintenance. The model to be presented contains the following notations.

**Decision Variables:**

\( d_{i,j} \): The allocated time of workstation \( i \) to batch \( j \) to accounting for setup, processing, starvation and blocking times of this batch. This is the time span from the point in time when setup of workstation \( i \) for batch \( j \) starts to the point when the last piece of the batch is moved to the subsequent buffer.

\( C_{i,j} \): Completion time of product \( j \) on workstation \( i \)

\( C_{m,n} \): Makespan, completion time of the last batch on the last workstation

\( N_j \): \( N_j \) is the batch size or the number of units of product \( j \) to produce. In cases when \( N_j \) is fixed in advance, it would be considered as a parameter and not a decision variable.

\( Util(i) \): Utilization of workstation \( i = \frac{\sum_j (N_j p_{ij} + s_{i,j})}{C_{m,n}} \)
Parameters:

- $b_i$: Size of buffer capacity between workstation $i$ and $(i+1)$. In cases when $b_i$ is not fixed in advance it would be considered as a decision variable and not a parameter.
- $p_{i,j}$: Processing time of one item of product $j$ on workstation $i$
- $s_{i,j}$: Setup time of workstation $i$ to process a batch of product $j$
- $Mn$: A maximum limit for makespan $C_{m,n}$. When this limit is not defined in advance then the constraints 4 to 8 would be eliminated.
- $T$: Manufacturing capacity. It is the maximum allowable time on any workstation for set-up, operation, as well as blocking and starvation of all $n$ batches together.

Indices:

- $m$: Number of workstations composing the production line
- $n$: Number of product types processed in the production line
- $i \in \{1,2,\ldots,m\}$: The index for workstations
- $j \in \{1,2,\ldots,n\}$: The index for product type
- $r$: A dummy index

The proposed model is an integer nonlinear programming model which may have a variety of objective functions including any of the following:

- **O1**: Maximize throughput $\sum N_j$, given buffer size and production capacity, $T$.
- **O2**: Maximize maintenance time, so maximize $\min_i \left( T - \sum_j d_{i,j} \right)$ or equivalently
  $$ \min_i \sum_j d_{i,j} $$
- **O3**: Minimize total production time $\sum \sum d_{i,j}$. This would be a helpful measure for costing purposes if the machines are leased.
- **O4**: Minimize makespan, $Mn$. It is important to know if the customer can be served within a particular timeframe.
- **O5**: Minimize the buffer size for a given peak demand $N_j \ \forall j$; capacity $T$; total production time $\sum \sum d_{i,j}$, and makespan $Mn$.

If we want to optimize more than one objective we may consider different levels of importance for the goals hence utilizing lexicographic goal programming. It should be noted that when we speak of the buffer size, basically we mean sum of all buffers or $\sum b_i$.
The feasible set of solutions is defined by the following set of constraints:

\[ d_{i,j} \geq N_jp_j + st_{i,j} \quad \forall i \& j \] (1)

\[ N_{j-k} \leq b_j \left[ d_{i,j-k} - st_{i,j-k} \right] \times \left[ -\sum_{r=0}^{k} d_{i,j-r} + \sum_{r=0}^{k} d_{i,j-r} - st_{i,j-k} + p_{i,j} + st_{i,j-k} \right]^{-1} \]

for \( i = 2, \ldots, m; \ j = 1, 2 \ldots n; \) and \( k = 0, 1, 2, \ldots, n-1 \) (2)

\[ N_j \geq b_j \left[ d_{i+1,j} - st_{i+1,j} \right] \times \left[ -\sum_{r=0}^{k} d_{i,j-r} - \sum_{r=0}^{k} d_{i,j-r} - p_{i,j-k} + st_{i+1,j-k} \right]^{-1} \]

for \( i = 1, 2, \ldots, m-1; \ j = 1, 2, \ldots, n; \) and \( k = 0, 1, 2, \ldots, n-1 \) (3)

\[ C_{i1} = d_{i1} \] (4)

\[ C_{i,j} \geq C_{i,j-1} + d_{i,j} \quad \forall i \geq 1, j \geq 2 \] (5)

\[ C_{i,1} \geq \sum_{r=1}^{i-1} \left( st_{r,1} + p_{r,1} \right) + d_{i,1} \quad \forall i \geq 2 \] (6)

\[ C_{i,j} \geq C_{i-1,j} + st_{i-1,j} + p_{i-1,j} + d_{i,j} \quad \forall i \geq 2, j \geq 2 \] (7)

\[ C_{m,n} \leq Mn \] (8)

\[ T \geq \sum_j d_{i,j} \quad \forall i \] (9)

\[ N_j \geq 0, \forall j; \quad d_{i,j} \geq 0, \forall i \& j; \quad b_j \geq 0, \forall i < m \]

It should be mentioned that the minimization of \( T \) does not guarantee the lowest inventory level. There are two approaches to cope with this problem. The first approach is basically of a two stage lexicographic goal programming. In the first stage, \( T \) is minimized. In the second stage, the minimum \( T \) becomes a constraint for the minimization of the buffer size. The second approach requires a penalty function of the buffer size, with a very small coefficient, to be included in the objective function (e.g. \( T + \varepsilon \sum b \)).

Constraint (1) states that the duration for which a workstation is occupied by a product is at least as much as the set up plus processing time because the duration, \( d_{i,j} \), accounts for set up, processing, blocking and starvation.

The size of the buffer has an impact on the dynamics of the in and outflow to the buffer. To avoid the filling up or exhausting buffers, Johri (1987) suggested two sets of constraints, which are called the input-side and the output-side constraints. These constraints have been modified and utilized in our proposed model in the form of constraints (2) and (3). Constraint (2)
states that the total amount of any product type produced at any workstation must not exceed the sum of the amount produced in the upstream workstation and the buffer between the two workstations; otherwise starvation would occur. Constraint (3) states that the total amount produced at any workstation must be at least equal to the sum of the amount produced in the downstream workstation and the buffer between the two workstations; otherwise blocking would result. Refer to Johri (1987) for detailed explanation of these constraints. In cases when $N_j$ is a decision variable, rather than being a parameter, constraints (2) and (3) become nonlinear.

Constraint (4) initializes the completion time of the first batch on the first workstation. Constraint (5) states that the completion time of batch $i$, on a workstation $i$, is at least equal the completion time of the previous batch plus the time allocated to this batch. Constraints (6) and (7) refer to our assumption regarding the item transfer environment explained earlier. Since each item is transferred to the next workstation individually, a batch of product type $j$ can start on workstation $i$ as soon as the first item of product $j$ is transferred to workstation $i$. Constraint (6) states that the completion time of batch $i$, on a workstation $i$, is at least equal to the time needed to setup and produce at least one unit on each of the upstream workstations plus the time needed in workstation $i$. Constraint (7) states that product $j$ cannot be completed on workstation $i$ until all items of product $j$ are processed on workstation $i$ which is started only after the previous product, product $(j - 1)$, is completed on workstation $(i - 1)$, workstation $(i - 1)$ is setup for product $j$ and at least one item of product $j$ is completed on workstation $(i - 1)$ and transferred to workstation $i$. Constraint (8) states that the last product must be completed on the last machine before the given makespan.

Constraint (9) states that the total time a workstation is occupied cannot be more than the capacity. The remaining are the non-negativity constraints of the decision variables.

Demonstration of Model’s Application

Our numerical example is intended to demonstrate how the model can be used to respond to various questions about the behaviour of the production system. Some typical questions are listed below:

- Given manufacturing capacity $T$, how does the $\sum N_j$ change with the changes in the buffer size?
- Given demand $N_j \forall j$, how does the $\max \sum d_{i,j}$ change with the changes in the buffer size?
- Given demand $N_j \forall j$, how does the $\sum \sum d_{i,j}$ change with the changes in the buffer size?
- Given demand $N_j \forall j$, how does the makespan, $Mn$ change with the changes in the buffer size?

The working of the model is demonstrated by means of a simple example of a production line composed of two workstations and processing two different products. This example will help us comparing our results to those already published in terms of the buffer’s contribution to the throughput.
Numerical Example:

This example demonstrates the use of the model to investigate the behaviour of the throughput of the production line. The throughput will be presented in terms of the sum of all batches processed per unit time, often an hour or a day. The contribution of the buffer to the throughput will be illustrated using a graphical representation where the production volume is plotted against the buffer size. In a different experiment, the impact of capacity $T$ on throughput is studied.

The production line under study is composed of two workstations with a buffer in between. The following parameter values define the scenario: (1) available capacity, $T = 28,800$ seconds (one shift of 8 hours); (2) the set-up time for each product in workstations 1 and 2 are: $st_{1,1} = st_{1,2} = 280$ seconds, and $st_{2,1} = st_{2,2}=210$ seconds; (3) the processing times per unit of product 1 in workstations 1 and 2 are $p_{1,1}=80$, and $p_{2,1}=30$; (4) the processing times per unit of product 2 in workstations 1 and 2 are $p_{1,2} = 44$ and $p_{2,2}= 75$; and (5) in the first set of experiments, the value of buffer capacity is changed from 1 to 125 and the daily throughput ($N = N_1+N_2$) is maximized (see Objective Function O1). Solving this integer NLP problem, we present the results in Figure 2.

**Figure 2**

**Daily Throughput vs. Buffer Sizes**

The optimal $N$, in Figure 2, was computed for a $T$ of 28,800 seconds. We can also test sensitivity of $N$ to the changes in $T$. The throughput does increase with increasing buffer size, however the marginal gain is decreasing and at some point the marginal gain becomes quite insignificant. At this point the capacity $T$ is the limiting factor to improve the throughput. When
the effects of the higher buffer size starts wearing down and the limiting effects of \( T \) starts showing up, trading starts to happen between the products that require more time per unit and the ones that are less time consuming to produce.

The findings explained above have managerial implications. The existence of the buffer makes possible the production of units that would have otherwise been left out. This is of relevance if there are minimal production requirements for each product. We can address the questions such as: how much buffer capacity would be required if a certain minimum production level of a product type \( j \) is mandatory. We can also establish the buffer size where no further increase in the throughput would be possible without additional manufacturing capacity.

The model permits us to examine the effects of manufacturing capacity, \( T \) on throughput and the required supporting buffer size. Figure 3 shows the effects of \( T \) on the optimal throughput and the required buffer size for the above problem. The results confirm our intuitive expectations that both the throughput and the required supporting buffer size increase with increasing capacity. A notable feature of Figure 3 is the apparent linear relation the optimal throughput and the buffer size have with capacity.

**Figure 3**

Throughput and the required buffer size as a function of capacity

Other results would be obtained from the model for objective function O2 or Minimize \( \max \sum_i d_{i,j} \). By considering the best throughput for products 1 and 2 as previously found (Figure 2) to be \( N_1=182 \) and \( N_2=122 \) at \( b=114 \), the optimal solution of the objective function O2 would be 20,488 seconds or 5.69 hours out of a total available 8 hours. So, 2.31 hours would be available for possible maintenance or other activities on workstation 1. Workstation 1 and 2's utilizations are: \( Util(1)= 0.711 \) and \( Util(2)= 0.526 \).
Considering the Objective Function O3, minimize \( \sum \sum d_{i,j} \): the total production time would be 35,518 seconds to make 182 units of product 1 and 122 of product 2. Solving for the optimal makespan (see objective function O4: Minimize \( Mn \)) given \( N_1 = 182 \) and \( N_2 = 122 \), the minimum makespan is equal to 35,289 seconds for an optimal buffer size = 115 units.

Given \( N_1 = 182 \) and \( N_2 = 122 \), a capacity of \( T = 28,800 \) seconds, total production time, \( \sum \sum d_{i,j} = 23,175 \) seconds and makespan, \( Mn = 29,900 \) seconds, solving for minimum buffer size or Objective Function O5, we get \( b = 5 \) units.

**Conclusion**

We have addressed here the problem of performance optimization in a highly automated production line composed of sequentially arranged machines with buffers in between. A nonlinear mathematical programming model was developed to describe the situation. A numerical example was presented and the various performance measures such as throughput, maintenance time, total production time, makespan and buffer sizes were demonstrated. The numerical example was also used to demonstrate how a manager may use the model to gain insight to the behaviour of the production line. Buffer size can contribute to increasing the output level but this contribution diminishes for larger buffer sizes. We also found that adjustments of the buffer size makes possible the production of some additional units that otherwise would not be produced under the optimal solution. Once the contributing effect of the buffer size diminishes the relative processing time advantages would favour the production of those products that require less processing time. While, based on the limited experimentation conducted in our research, we cannot extrapolate these findings to a general scenario; nevertheless it was quite enlightening to find such fact by following the methodology as indicated earlier. The implication for production managers is in establishing the most economical use of buffers to increase the total throughput. This would result in a better inventory management and a better allocation of buffer capacity to different buffer locations.

**Acknowledgement**

The authors would like to thank NSERC for financial supports.
References


