

SIMULATION–OPTIMIZATION OF FLOW LINES: AN LP-BASED BOUNDING APPROACH

Arianna Alfieri

Dipartimento di Sistemi di
Produzione ed Economia dell’Azienda
Politecnico di Torino
Torino, Italy

Andrea Matta

Dipartimento di Meccanica
Politecnico di Milano
Milano, Italy

ABSTRACT

Mathematical programming representation has been recently used to describe the behavior of discrete event systems as well as their formal properties. This paper proposes approximate mathematical programming models for the simulation–optimization of flow lines with finite buffer capacities. The approximation exploits the concept of time buffer, modeled as a constraint that put into a temporal relationship the completion times of two jobs in a sample path. The main advantage of the proposed formulation is that it preserves its linearity even when used for buffer optimization in multistage flow lines. The solution of the approximate model can be used to obtain bounds on the variables of the exact model, to reduce its feasible region and hence the computation time to find the optimal buffer allocation for the line.

1 INTRODUCTION

Simulation is one of the most popular techniques to study the behavior of production systems in situations in which it is not possible to define mathematical expressions for describing the system behavior. Simulation allows to predict the behavior of production systems, given some defined working conditions and system parameters. Optimization, instead, allows to determine the best working conditions or system parameters, but it can be used only when a mathematical representation of the system behavior is available and does not allow to directly predict system performance. Optimization via simulation consists in applying iteratively two detached models until an optimality condition is reached: a simulation model for predicting the system performance and an optimization model for generating and selecting potential optimal solutions (Fu 2002) (Fu, Glover, and April 2005).

The recent works (Chan and Schruben 2003) (Chan and Schruben 2008) on mathematical programming for modeling discrete event systems pave the way to the development of new formulations that can be used for optimization while the system performance measures are calculated at the same time. In (Matta 2008) the buffer allocation problem in flow lines has been modeled with both exact and approximate *sim–opt* models. The exact optimization model is a MILP model whose complexity increases as the number of simulated parts and the length of the line also increase. Thus, such a model has no practical application in complex real cases due to the computational effort it would require. The approximate optimization model, instead, is an LP model with a computational complexity affordable even if long lines are considered. In fact, the usage of the approximate model can allow the analysts to work mainly in the linear domain, thus taking the computational effort low. However, it is not described what the approximate formulation effectively represents and its relation with the exact one.

This paper deals with approximate modeling of queuing systems by mathematical programming formulations, exactly with the goal of finding the relations between the proposed approximate model and the exact model proposed in (Chan and Schruben 2008). In particular, we develop two formulations for the simulation–optimization of a specific class of queuing systems: flow lines with finite buffer capacity. The first model is based on the standard concept of *space buffer*. The second model is an LP that exploits the

concept of *time buffer* as an alternative way of modeling queues in discrete event systems. The relations between exact and approximate LP formulation are formally studied in the paper. An interesting result is that it is possible to move from the approximate model with time buffers to the exact one with space buffer. When the model is used for system optimization, the relations between exact and approximate LP formulations allow the analysts to identify the most interesting subset of feasible buffer capacity for each stage of the line. This result is particularly important since the exact model for the buffer allocation problem is, differently from performance evaluation exact models, a MILP model. In fact, the buffer capacity subsets identified by the approximate LP model can be used to reduce the feasible region (and hence reducing the computational burden) in solving the problem to find the optimal buffer space allocation in transfer lines.

The contribution of the paper resides in the strong characterization of a new concept of buffer (the *time buffer* concept) and in the definition of formal relations between exact and approximate formulations that are exploited to develop an effective bounding procedure. In the literature related to optimization of flow lines there is no method to derive the bounds for the optimal buffer capacities in flow lines composed of machines with generally distributed processing times.

The plan for this paper is the following. In Section 2 we formally define the analyzed system and the used notation. Section 3 and Section 4, respectively, present the exact and approximate LP formulation to simulate and optimize the flow line. The time buffer concept is also introduced in Section 4. In Section 5 the relations between exact and approximate formulations are discussed while the bounding procedure is described in Section 6. Numerical results on randomly generated instances are reported in Section 7. Section 8 concludes the paper.

2 ASSUMPTIONS AND NOTATION

We consider a flow line with J single-machine stages, on which N identical items have to be processed. The processing sequence is known in advance, i.e., no scheduling problem needs to be solved. In particular, item i is processed before item $i + 1$ on each machine of the line. The arrival time of each item, denoted by a_i , and the processing times of each item on each machine j , denoted by t_{ij} , are assumed known in advance from a random sampling or from a specific sample path. Since no scheduling decision is considered, $a_i \leq a_{i+1}$ for all items i . After having been processed by the first machine, parts proceed to the second machine, then to the third and so forth until the last operation is performed at the last machine; finally parts leave the system. Starting and completion times of part i at machine j are denoted with x_{ij} and y_{ij} respectively.

Each machine j , excluding the first, has an incoming buffer B_{j-1} . The capacity of buffer B_j is denoted with c_j , and this includes also the part under processing at the machine $j + 1$. Part i has to wait in buffer B_{j-1} if machine j is busy in processing another part k (with $k < i$). The blocking before service control rule is assumed for machines (Dallery and Gershwin 1992). Machines are perfectly reliable, i.e., no failure possibility is considered (which is the same as assuming failure possibilities with negligible repair times). Transportation times are considered negligible or already included in machining times. Finally, for sake of simplicity, the last machine is never blocked, thus parts completing their processing can always leave the system.

The performance measure considered in this paper is the mean production rate of the line (denoted with θ), defined as the average number of parts produced by the line in a time window.

3 EXACT REPRESENTATIONS

3.1 Formulation for Exact Simulation of Flow Lines

A linear programming (LP) model is now described to simulate a generic open flow line with J machines separated by buffers with finite capacity. This simplified version of LP model can be obtained from (Chan

and Schruben 2003):

$$\begin{aligned}
 & \min && \sum_{i=1}^N \sum_{j=1}^J y_{ij} && (1) \\
 & \text{s.t.} && && \\
 & && y_{i1} \geq a_i + t_{i1} && \forall i && (2) \\
 & && y_{i+1,j} - y_{ij} \geq t_{i+1,j} && \forall j, i = 1, \dots, N-1 && (3) \\
 & && y_{i,j+1} - y_{ij} \geq t_{i,j+1} && \forall i, j = 1, \dots, J-1 && (4) \\
 & && y_{i+c_j,j} - y_{i,j+1} \geq t_{i+c_j,j} && i = 1, \dots, N - c_j, && \\
 & && && j = 1, \dots, J-1 && (5)
 \end{aligned}$$

Constraints (2) simply impose that the service at the first machine cannot start before the arrival time of the same part at the system plus its processing time on the first machine. Constraints (3) mean that a machine cannot process two consecutive parts at the same time. Constraints (4) state that a part cannot be contemporary processed by two different machines at the same time. Constraints (5) impose that a part cannot leave a machine if the immediate downstream buffer is full. The solution of the linear problem provides the optimal values for decision variables y . Finishing times can assume only positive values in the real domain because the arrival times are nonnegative input parameters.

The problem solution corresponds to the dynamic behavior of the discrete event system, i.e., the optimal values are exactly the finishing time events of machining operations in a real or simulated system that has the same sequence of parts, the same arrival events and the same processing times. See the work of Chan and Schruben for more details (Chan and Schruben 2008).

A simulated sample path of the exact model is unequivocally determined by the tuple $P = (S, E)$, where S and E represent the flow line configuration and the part characteristics respectively. In detail, S is a $J - 1$ -dimensional vector containing the space buffer capacities, E is a tuple containing the vector of arrival times and the matrix of processing times at each machine for each part in the analyzed sample path. The mean production rate related to the sample path can be calculated after the LP model has been solved:

$$\theta = \frac{N}{y_{NJ}}. \tag{6}$$

3.2 Formulation for Exact Buffer Capacity Optimization of Flow Lines

The optimization problem (i.e., the problem of choosing the optimal capacity for each buffer) is represented by the following MILP model (Matta 2008):

$$\begin{aligned} \min \quad & \sum_{j=1}^{J-1} e_j \sum_{k=L_j}^{U_j} z_{jk} k \end{aligned} \tag{7}$$

s.t.

$$\begin{aligned} & (2) - (4) \\ y_{i+k,j} - y_{i,j+1} & \geq t_{i+k,j} z_{jk} - (1 - z_{jk}) M \quad j = 1, \dots, J - 1, \\ & \quad \quad \quad i = 1, \dots, N - k, \\ & \quad \quad \quad \forall k \end{aligned} \tag{8}$$

$$\sum_{k=L_j}^{U_j} z_{jk} = 1 \quad \forall j \tag{9}$$

$$y_{NJ} - y_{dJ} \leq \frac{N - d}{\theta^*} \tag{10}$$

where z_{jk} is a binary variable equal to one if capacity k (with $k = L_j, \dots, U_j$) is assigned to queue B_j and e_j is the weight (i.e., cost) of unit capacity at queue B_j ; values L_j and U_j are the bounds defined by the analyst of the optimization problem for the j -th buffer. The objective function is to minimize the weighted sum of the buffer capacities. Constraint (10) imposes that the mean throughput must be greater than or equal to a minimum value θ^* . If $z_{jk} = 1$, then all the constraints related to queue B_j with assigned capacity equal to k are activated; otherwise, the constraint is made redundant by subtracting from the right-hand side the big M (8). Index d represents the end of the system warm-up, identifiable with well-known techniques. Finally, only one capacity k must be chosen for each buffer B_j as imposed by equation (9).

Notice that the only differences between the simulation and the optimization model are the objective function and the constraints (8) and (9) that substitute constraints (5).

All these constraints refer to the buffer capacity. While in the simulation model, buffer capacity is a parameter (c_j), in the optimization model it is a decision variable (z_{jk}). Moreover, in the optimization model it is required a constraint on the minimum level of throughput, otherwise the system would select zero buffer capacity for each stage.

Finally, notice that the larger is the number of available buffer capacity (i.e., the larger the difference $U_j - L_j$), the larger is the number of constraints (8) and hence the more difficult will be to optimally solve the problem. Without any knowledge about the system, the only possibility not to “cut” the optimal solution out from the feasible region is to set $L_j = 1$ and $U_j = N$, but in real cases this choice results in a computationally infeasible problem.

The approximate models presented in the next section will help to reduce the feasible region.

4 APPROXIMATE REPRESENTATIONS

4.1 Time Buffer

The exact formulation uses the concept of *space buffer* to simulate the behavior of the system. The space buffer is what is commonly thought of when speaking of buffer capacity, i.e., an available space, between two adjacent machines, where items can wait to be processed.

An alternative way of modeling the possibility for items to wait in a queue, before being processed on a machine, is the use of the *time buffer*. A time buffer can be considered as the time length an item can be started “in advance” on a machine before the successive machine is available for processing. Given two

parts a and b to be processed in the sequence $a \rightarrow b$ (i.e., a precedes b), a time buffer of capacity s_j in front of machine $j + 1$ forces the starting time of job b at machine j (denoted with x_{bj}) and finishing time of job a on machine $j + 1$ (denoted with $y_{a,j+1}$) in the following way:

$$x_{bj} \geq y_{a,j+1} - s_j$$

The above equation simply says that job b at machine j can start s_j time units before the completion of job a at machine $j + 1$. When the time buffer is null, the two adjacent machines are perfectly synchronized. On the other hand, two machines are decoupled when there is an infinite time buffer between them. The main difference between *space* and *time* buffer is the fact that a space buffer of capacity m is always able to accommodate m items, independently from the system conditions. Also, the time spent by parts stored in the space buffer is not limited. On the contrary, the number of parts that, for instance, time buffer B_j can store is not known a priori; in fact, it depends on the part flow arriving from machine $j - 1$ in a certain time window (equal to the time buffer) just before machine j ends its processing. Obviously, the time spent by parts in the time buffer is limited by its capacity.

Finally, notice that the time buffer concept introduced in this section is different from that adopted in manual assembly lines, which is mainly related to dimensioning the speed of the conveyor between two adjacent machines. The proposed time buffer is also different from the slack time in PERT graphs, which can be defined as the time available between the estimated completion time of the job and its due date.

4.2 Formulation for Approximate Simulation of Flow Lines

The approximate mathematical representation of a flow line that considers the time buffer concept is now introduced.

$$\begin{aligned}
 & \min \\
 & \quad \sum_{i=1}^N \sum_{j=1}^J y_{ij} \\
 & \text{s.t.} \\
 & \quad (2) - (4) \\
 & \quad y_{i+k,j} - y_{i,j+1} \geq t_{i+k,j} - s_{kj} \quad i = 1, \dots, N - 1, \\
 & \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad j = 1, \dots, J - 1, k = 1, \dots, c_j
 \end{aligned} \tag{11}$$

In equations (11) parameter $s_{kj} \in \Re$ represents the time buffer capacity between machines j and $j + 1$ constraining the finishing times of parts $i + k$ and i . Such formulation presents a specific time buffer capacity between two completion times at two adjacent machines. Time buffers constrain completion times distant at maximum lag c_j .

The only set of constraints that differs from the exact formulation is the one related with the buffer capacity. Indeed, the space buffer constraints (5) are replaced by the time buffer constraints (11).

A simulated sample path of the approximate model is unequivocally determined by the tuple $P' = (S', E')$, where S' and E' represent the flow line configuration and the part characteristics respectively. Specifically, S' is the matrix s containing time buffer capacities s_{kj} ; E' is a tuple containing the vector of arrival times and the matrix of processing times at each machine for each part in the analyzed sample path. After this LP model has been solved, an approximation of the mean production rate of the sample path can be calculated using equation (6).

It is well known in the literature that the throughput is a monotonic function of the space buffer. The same result holds for the time buffer model, as defined in the following property.

Property 1 Let $P' = (S', E')$ be an approximate formulation of a flow line sample path. The resulting throughput of the approximate model is a monotonic function of s_{kj} , i.e., if the values of s_{kj} increase for

all k and j , the throughput cannot decrease and if the values of s_{kj} decrease for all k and j , the throughput cannot increase.

Proof. In the approximate model, if s_{kj} increases for all k and j , constraints (11) allow finishing times $y_{i,j+1}$ to decrease. Since this is applicable to each j and k , also y_{NJ} is allowed to decrease. Hence, by equation (6), throughput θ cannot decrease.

On the other way around, if s_{kj} decrease for all k and j , constraints (11) force finishing times $y_{i,j+1}$ to increase. Since this is applicable to each j and k , also y_{NJ} is forced to increase. Hence, by equation (6), throughput θ cannot increase. Hence, the throughput is a monotonic function of the time buffer. \square

4.3 Formulation for Approximate Buffer Capacity Optimization of Flow Lines

As in the case of the space buffer formulation, by changing the approximate simulation model devised above, it is possible to obtain the approximate model for the optimization problem:

$$\begin{aligned} \min \quad & \sum_{j=1}^{J-1} e_j \sum_k s_{kj} \end{aligned} \tag{12}$$

s.t.

$$\begin{aligned} & (2) - (4) \\ y_{i+k,j} - y_{i,j+1} & \geq t_{i+k,j} - s_{kj} \quad i = 1, \dots, N-1, \\ & j = 1, \dots, J-1, k = 1, \dots, K_j \end{aligned} \tag{13}$$

$$y_{NJ} - y_{dJ} \leq \frac{N-d}{\theta^*}$$

Equation (12) is the objective function, i.e., the minimization of the overall buffer costs. Again, the objective function is to minimize the weighted sum of the time buffer capacities. Constraints (13) prevent a part from leaving a machine if the immediate downstream time buffer is full. Index k is defined between 1 and an upper value K_j defined by the analyst. Parameter K_j cannot be larger than N obviously. As in the case of the exact optimization model, a throughput constraint is necessary to prevent the model to reach the “optimal” solution with zero cost, i.e., in which s variables are all null. The solution of this model is the array of time buffer capacities that minimize the total cost in equation (12) under the throughput constraint. Notice that if s_{kj} is positive for some k , it means that a buffer B_j , of some capacity, is needed. Starting from this observation, we will show in the next section how it is possible to obtain from the approximate optimal solution bounds L_j and U_j to reduce the feasible region in the space buffer domain.

5 RELATIONS BETWEEN SPACE AND TIME BUFFER MODELS

The aim of this section is to present some structural results that allow the analyst to map an approximate formulation to a set of exact formulations. Before illustrating the relations between exact and approximate formulations, notice that the performance measure θ cannot decrease if the feasible region of the problem is enlarged. This is proved in the following lemma.

Lemma 1 Let $P = (S, E)$ be an exact (or approximate) LP formulation of a flow line sample path. If the feasible region Ω , defined by constraints (1) - (5) (or by constraints (1) - (4), (11)) is enlarged to Ω' by removing some of the constraints, the mean production rate θ' on Ω' is such that $\theta' \geq \theta$.

Proof. If some constraints are removed, variables y are able to reach smaller values (we are dealing with a minimization problem). Since y_{NJ} is the finishing time of the last part on the last machine, even if it does not directly decrease, it cannot increase if other y have reached possibly smaller values. Hence y'_{NJ} on Ω' cannot be bigger than y_{NJ} on Ω , which implies that $\theta' \geq \theta$. \square

Analogously, it can be proved that, if Ω is reduced to Ω' by adding some constraints, the mean production rate cannot increase, i.e., $\theta' \leq \theta$.

Theorem 1 Let $P = (S, E)$ be an exact formulation of a flow line sample path, and let $P' = (S', E')$ be an approximate formulation of the same flow line, with $E = E'$. If the following inequalities are valid:

$$s_{kj} > x_{i+c_j,j} - x_{i+k,j} \quad k = 1, \dots, c_j, i = 1, \dots, N - c_j, \\ j = 1, \dots, J - 1 \quad (14)$$

then $\theta' \geq \theta$.

Analogously, if the following inequalities are valid:

$$s_{kj} < x_{i+c_j,j} - x_{i+k,j} \quad k = 1, \dots, c_j, i = 1, \dots, N - c_j, \\ j = 1, \dots, J - 1 \quad (15)$$

then $\theta' \leq \theta$.

Proof. It is easy to demonstrate that when the inequalities (14) are valid, the feasible region of the approximate formulation is larger than and contains that of the exact model. Constraints (5) and (11) can be rewritten as:

$$y_{i,j+1} \leq x_{i+c_j,j} \quad (16)$$

$$y_{i,j+1} \leq x_{i+k,j} + s_{kj}. \quad (17)$$

If the condition in (14) is true for all the possible values of k and j :

$$s_{kj} + x_{i+k,j} > x_{i+c_j,j}.$$

Using the above inequality, it is easy to demonstrate that equation (16) constrains more than equation (17). Thus, by Lemma 1 we can conclude that the production rate estimated by the approximate model is greater than or equal to that estimated by the exact model.

The second part of the theorem can be proved in a similar way, hence we omit its demonstration. \square

The following theorem allows us to relate an approximate model having a certain time buffer configuration with an exact model having some defined performance.

Theorem 2 Let $P' = (S', E')$ be an approximate formulation of a flow line sample path, and $\mathbf{s} = [s_{kj}]$ be the matrix of the k -lag time buffer capacities between machines j and $j + 1$. Let also y_{ij} , x_{ij} and θ' be the result of the approximate model. If there exists a value of $c_j^u = 1, \dots, N - 1$ for which the following inequalities are satisfied:

$$s_{kj} < x_{i+c_j,j} - x_{i+k,j} \quad i = 1, \dots, N - c_j^u, \\ j = 1, \dots, J - 1, 1 < k \leq c_j^u \quad (18)$$

and a value of $c_j^l = 1, \dots, N - 1$ for which the following inequalities are satisfied:

$$s_{kj} > x_{i+c_j,j} - x_{i+k,j} \quad i = 1, \dots, N - c_j^l, \\ j = 1, \dots, J - 1, 1 < k \leq c_j^l \quad (19)$$

then there is at least one space buffer configuration c_j in the set $c_j^l \leq c_j \leq c_j^u$ having a throughput θ such that $\theta^l \leq \theta \leq \theta^u$.

Proof. Given a value of c_j , if condition (18) or condition (19) are valid then Theorem 1 can be applied to derive $\theta \leq \theta'$ or $\theta \geq \theta'$. \square

This result simply states that the approximate formulation of a flow line with a certain time buffer configuration overestimates or underestimates the throughput of a flow line modeled with space buffers having capacities equal to c_j , being invariant arrival and processing times (i.e., $E = E'$). Notice that the conditions in (18) and (19) can be true for different values of c_j .

Once s_{kj} have been fixed, also the mean production rate of the approximate flow line is given. By applying Theorem 2, the bounding space buffer capacities to have the same performance, c_j^l and c_j^u , can be then determined. These values can be used in the space buffer MILP optimization model to reduce the problem search domain, i.e., to help the computational efficiency of solution algorithms. Notice, in fact, that c_j^l and c_j^u represent the bounds L_j and U_j used in Section 3.2.

6 BOUNDING PROCEDURE

In this section we describe a general procedure for the optimization of buffer capacities in flow lines.

1. *Initialization:* generate randomly arrival (a_i) and processing (t_{ij}) times for all the N parts of the sample path. Set the time buffer capacities to the upper feasible value (K_j , with $j = 1, \dots, J-1$).
2. *Approximate optimization:* solve the approximate optimization model (Section 4.3) on the sample path generated at step 1. Collect the time buffer capacity \mathbf{s}^* from the optimal solution.
3. *Approximate simulation:* solve the approximate simulation model (Section 4.2) using \mathbf{s}^* as an input parameter on the sample path generated at step 1. Collect variables x_{ij} from the model solution ($i = 1, \dots, N$; $j = 1, \dots, J-1$).
4. *Bounding:* apply equations (18) and (19) to calculate lower and upper bounds c_j^l and c_j^u for each buffer capacity ($j = 1, \dots, J-1$). In details, for each j , the parameter c_j in equation (18) is fixed to 1 and inequality $s_{kj} < x_{i+1,j} - x_{i+k,j}$ is checked for each feasible i and k . If the inequality is verified for each i and k , hence $c_j^u = 1$. Otherwise, if it is not verified for some i or k , c_j is increased by 1 and inequality $s_{kj} < x_{i+2,j} - x_{i+k,j}$ is checked, and so on until a value of c_j for which $s_{kj} < x_{i+c_j,j} - x_{i+k,j}$ is verified for each feasible i and k is found. The same is done to find the lower bound c_j^l , with the only difference that equation (19) is used for each j and each value of c_j .

At the end of this procedure an optimization algorithm can be applied to the exact optimization model (Section 3.2) for finding out the optimal solution in the feasible area restricted by the calculated bounds. The use of the bounds should lead to a reduction of the computational time necessary to solve the optimal buffer capacity allocation problem.

7 NUMERICAL TESTS

Numerical tests on randomly generated instances were carried out to evaluate the strength of the bounds developed in the previous section. We executed three sets of experiments, considering different simulation settings in terms of number of machines in the line, saturation of the line and simulation type (terminating or steady-state).

The first set of experiments deals with a terminating simulation in a three machine flow line. We simulated the processing of 500 parts all available at time zero (i.e. saturated line with $a_i = 0, \forall i$). Notice that the *availability at time zero* assumption does not reduce the generality of the experiment; similar results can be obtained by generating arrivals so that $a_i > 0$. The computational times on the three machines have been assumed exponentially distributed with average values of 0.4, 0.5 and 0.2 time units respectively on the first, second and third machine of the line. The bottleneck is clearly the second machine and the maximum achievable throughput θ is therefore 2 parts per time unit. Using the approximate optimization model we found the optimal time buffers s_{kj}^* that allow to reach an a priori given value of $\theta^* = 1.7$ (Step

2 of the bounding procedure).

The values of s_{kj}^* have been used in the approximate simulation model presented in Section 4.2 (Step 3 of the bounding procedure). Theorem 2 has then been applied to find c_j^l and c_j^u from the model solution (Step 4 of the bounding procedure). Finally, the exact simulation model presented in Section 3.1 is solved first with c_j^l for each j and then with c_j^u for each j , computing the respective throughput θ^l and θ^u . The above described experiment has been repeated after having changed the values of s_{kj} in the approximate simulation model, i.e., by repeating Step 3 of the bounding procedure with different values of time buffer capacities. In particular, we decreased all s_{kj}^* to $0.5s_{kj}^*$ and increased to $2s_{kj}^*$.

We experimented with 10 different sample paths randomly generated. The results are reported in Table 1. In such tables, *Run* refers to the sample path. Columns s_1 and s_2 contain the total time buffer of buffer B_1 (between the first and the second machine) and B_2 (between the second and the third machine). The total time buffer is defined as $\sum_k s_{kj}$. Columns c_1^l and c_1^u correspond to the values of c^l and c^u for buffer B_1 , while c_2^l and c_2^u have the same meaning for buffer B_2 . The last two columns, θ^l and θ^u , are the throughput given by the exact simulation model when instantiated with the same sample path as the approximate simulation model and the c_j of both buffers respectively at the *lower* and *upper* bound values.

The range on space buffer values, i.e., $c_j^u - c_j^l$, seem to be not so tight. Notice, however, that the possible space buffer values range between 1 (no buffer) till N (buffer able to accommodate all the parts at the same time). Hence, the bounds determined by using Theorem 2 allow, in the worst case (Table 1, Run 5, Buffer B_1) a reduction of the feasible values of about 87%.

The second set of experiments considers also a three machine flow line, with exponentially distributed processing time and all parts available at time zero, with a steady-state simulation. In particular, we simulated 10,000 parts, with the first 2,000 parts corresponding to the warm-up period. The average processing time of the three machines are respectively 0.1429, 0.1429 and 0.1667 time units. The maximum achievable throughput is therefore 6 parts per time unit.

We applied the bounding procedure using a desired throughput $\theta^* = 5.4$ (i.e., 90% of the maximum achievable) and, also in this case, we considered 10 different random sample paths.

The results of the second set of experiments are reported in Tables 2 and 3. Table 2 reports the total time buffer capacities and the throughput corresponding to the solution of the exact simulation model instantiated with the same sample path as the approximate one and with c_j values to the upper bound for both buffers (column θ^u), to the lower bound for both buffers (column θ^l), the first buffer to the upper and the second to the lower (column θ^{ul}), the first buffer to the lower and the second to the upper (column θ^{lu}). Table 3 reports the calculated bounds and the optimal space buffer configuration that minimizes the total allocated buffer capacity under the constraint $\theta \geq 5.4$. Optimal solutions were obtained by solving the MILP model in Section 3.2. These optimal values, by Theorem 2, are always between the values c_l and c_u identified from the solution of the approximate model.

In the last set of experiments we considered a two-stage system to show how the quality of the bounds changes as a function of the system throughput. The two stages have the same exponential distribution with mean equal to 0.1 time units. Experiments were executed on a fixed sample path using different values of the corresponding total time buffer. The results are reported in Figure 1. Also in this case, it is possible to notice that increasing the time buffer allows to reach higher values of throughput, as stated by Property 1. Moreover, the range of throughput given by the solution of the exact model instantiated with c^l and c^u , computed as $(\theta^u - \theta^l)$, first increases and hence decreases, being quite tight for high level of system saturation. This behavior is the same as observed in the three machine case. Indeed, the first increase in the throughput range is due to the fact that, when the saturation level is low, increasing s_{kj} leads to an increase of c_j^u , while c_j^l tends to remain fixed. Increasing the saturation level further towards the 100%, not only c_j^u increases, but also c_j^l does, thus reducing the throughput range.

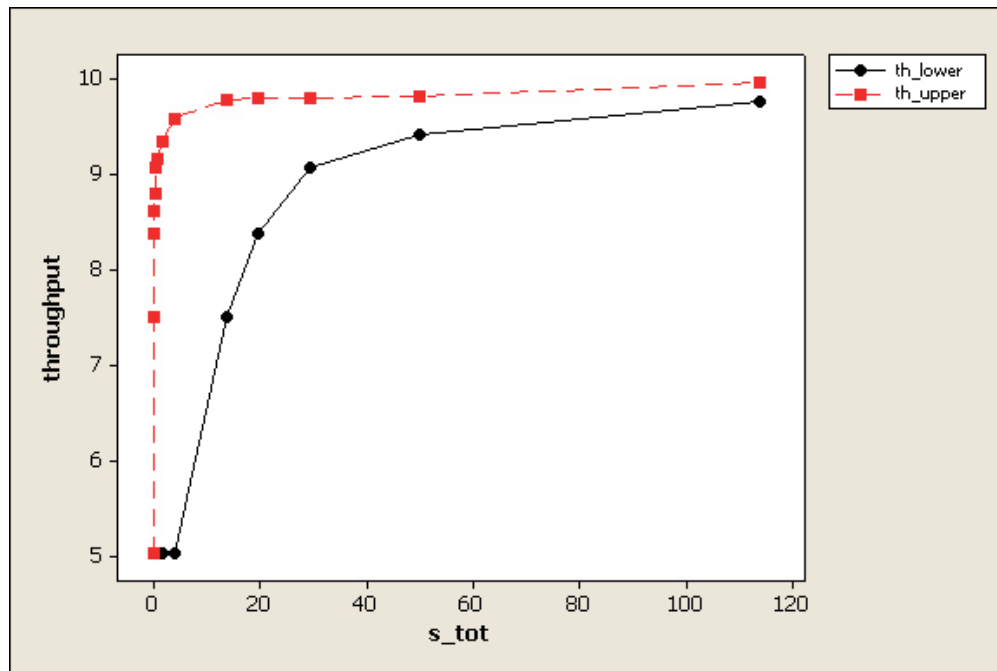


Figure 1: Lower and upper bound on two-machine flow line throughput.

Therefore, we can conclude that the approximation seems to improve in the high saturated areas, i.e., in those areas in which flow lines generally operate.

The time for solving the approximate simulation and optimization models in Steps 2 and 3 is in the order of seconds (first and third experiment) or minutes (second experiment) on an Intel CoreTM2 Duo 2.53GHz 2.99 GB RAM.

8 CONCLUSION

In this paper we considered approximate and exact models to evaluate and optimize the performance of a flow line with finite buffer capacity. The exact model has been developed based on the well-known concept of *space buffer*, while in the approximate model we introduced the new concept of *time buffer* as an alternative way of modeling space buffer.

The relations between the two models have been formally studied to allow the analyst to move from one model to the other, i.e., to optimize the model in the approximate domain and use the optimization results to identify the most promising areas in the exact domain in which to intensify the search of the optimal solution to buffer allocation problem.

The results of numerical tests on random generated instances show the efficacy of the bounds given by the approximate model to reduce the feasible region for the space buffer capacity in the exact optimization model. A further step will be the assessment of the computational time saving that derives from the application of the bounding procedure in simulation–optimization algorithms such as OPTQUEST and COMPASS (Hong and Nelson 2006). Furthermore, in this paper only a sample path is considered, however the proposed approximate model can be exploited to define bounds that take into account several replications under the stochastic programming framework. This subject is an ongoing research.

Table 1: Results for terminating simulation ($\theta^* = 1.7$).

Run	s_1	s_2	c_1^l	c_1^u	c_2^l	c_2^u	θ^l	θ^u
1	2.290	0.805	2	6	1	4	1.2247	1.8220
	4.580	1.610	2	7	1	6	1.2247	1.8687
	9.160	3.220	3	12	1	9	1.3197	1.8952
2	2.140	0.740	2	5	1	3	1.2373	1.7388
	4.280	1.480	2	8	1	6	1.2373	1.8593
	8.560	2.960	2	12	1	8	1.2373	1.8772
3	0.875	0.350	1	4	1	3	1.0920	1.8474
	1.750	0.700	1	6	1	4	1.0920	1.9675
	3.500	1.400	1	10	1	7	1.0920	2.0200
4	1.355	0.645	1	6	1	4	1.0552	1.8488
	2.710	1.290	1	8	1	5	1.0552	1.9070
	5.420	2.580	2	12	1	11	1.2755	1.9490
5	1.425	0.390	1	6	1	3	1.0361	1.8575
	2.850	0.780	1	9	1	5	1.0361	1.9734
	5.700	1.560	1	13	1	7	1.0361	2.0100
6	2.845	0.990	1	7	1	6	0.9682	1.7497
	5.690	1.980	1	10	1	7	0.9682	1.7904
	11.380	3.960	3	14	1	10	1.2724	1.8112
7	1.935	0.745	1	6	1	5	1.0062	1.8166
	3.870	1.490	1	8	1	6	1.0062	1.8615
	7.740	2.980	2	15	1	10	1.2456	1.9364
8	1.285	0.445	1	6	1	4	1.0535	1.9287
	2.570	0.890	1	8	1	5	1.0535	1.9982
	5.140	1.780	1	12	1	7	1.0535	2.0400
9	2.540	0.675	1	6	1	5	1.0114	1.7152
	5.080	1.350	1	10	1	7	1.0114	1.8855
	10.160	2.700	3	15	1	9	1.3227	1.9271
10	2.000	0.845	1	8	1	5	1.0146	1.8095
	4.000	1.690	1	10	1	9	1.0146	1.8328
	8.000	3.380	1	12	1	11	1.0146	1.8415

Table 2: Results for steady-state simulation: time buffers and performance.

Run	s_1	s_2	θ^l	θ^{ul}	θ^{lu}	θ^u
1	4.1210	6.2251	2.6353	3.1794	3.4955	5.7852
2	3.1028	4.1729	2.6706	3.2364	3.4992	5.9170
3	3.6635	5.2830	2.6346	3.1844	3.4588	5.8771
4	3.5194	5.0957	2.6450	3.1967	3.4971	5.8034
5	2.9703	4.2171	2.6799	3.2385	3.5120	5.9098
6	2.5699	3.9818	2.6897	3.2533	3.5249	5.8273
7	3.3947	4.2498	2.6629	3.2405	3.5144	5.8600
8	3.1628	3.9666	2.6537	3.2521	3.4659	5.9779
9	3.3936	5.2277	2.6440	3.2190	3.4986	5.8327
10	2.8097	3.9853	2.6791	3.2659	3.5142	5.9729

Table 3: Results for steady-state simulation: bounds and optimal values.

Run	c_1^l	c_1^u	c_2^l	c_2^u	c_1^*	c_2^*
1	1	17	1	20	6	9
2	1	14	1	15	6	7
3	1	17	1	17	6	9
4	1	17	1	17	6	8
5	1	17	1	16	5	8
6	1	14	1	9	5	7
7	1	15	1	16	6	8
8	1	15	1	16	6	7
9	1	16	1	17	6	8
10	1	13	1	15	6	7

REFERENCES

- Chan, W., and L. Schruben. 2008. "Optimization models of Discrete-Event System Dynamics". *Operations Research* 56 (5): 1218–1237.
- Chan, W. K., and L. W. Schruben. 2003, December. "Properties of Discrete Event Systems from their Mathematical Programming Representations". In *Proceedings of the 2003 Winter Simulation Conference*, edited by S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 496–502. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Dallery, Y., and S. B. Gershwin. 1992. "Manufacturing Flow Line Systems: A Review of Models and Analytical Results". *Queueing Systems Theory and Applications, Special Issue on Queueing Models of Manufacturing Systems* 12 (1-2): 3–94.
- Fu, M. 2002. "Optimization for Simulation: Theory vs. Practice". *Journal on Computing* 14 (3): 192–215.
- Fu, M., F. Glover, and J. April. 2005, December. "Simulation optimization: a review, new developments, and applications". In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 83–95. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Hong, L. J., and B. L. Nelson. 2006. "Discrete optimization via simulation using COMPASS". *Operations Research* 54:115–129.
- Matta, A. 2008, December. "Simulation Optimization with Mathematical Programming Representation Of Discrete Event Systems". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, T. Jefferson, and J. W. Fowler, 1393–1400. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

ARIANNA ALFIERI is Assistant Professor at Politecnico di Torino, where she currently teaches production planning and control and logistic system simulation. Her research area includes scheduling and planning in production and transportation systems. Her email address is arianna.alfieri@polito.it.

ANDREA MATTA is Associate Professor at Politecnico di Milano, where he currently teaches manufacturing and integrated production systems. His research area includes analysis, design and management of production and service systems. His email address is andrea.matta@polimi.it.