Least In-Sequence Probability Heuristic for Mixed-Volume Production Lines

OLEG Y. GUSIKHIN
University of Michigan Business School,
Ann Arbor, MI 48109-1234
E-mail: OGusikhi@umich.edu

RAHUL CAPRIHAN
Faculty of Engineering,
Dayalbagh Educational Institute,
Dayalbagh, Agra. India 282 005
Phone: +91 (562) 220428
Email: Rahulcap@giasdl01.vsnl.net.in
Fax:+91 (562) 220428

KATHRYN E. STECKE
University of Michigan Business School,
Ann Arbor, MI 48109-1234
Phone: (734)763-0485
E-mail: KStecke@umich.edu

- October, 1999

Least In-Sequence Probability Heuristic for Mixed-Volume Production Lines

ABSTRACT

This paper focuses on the sequencing aspects of a flexible assembly system (FAS) functioning in a build-to-order environment. An important feature of the FAS considered in this study is that the demand sequence of part types is known and fixed for a given period of time. Further, the different part types that constitute the demand sequence can have different frequencies of occurrence in a range specified from low to high. Interestingly, for different sections of the demand sequence, the frequency of occurrence for a given part type is not fixed. We exploit this property of the demand sequence in the development of the least in-sequence probability (LISP) heuristic rule. The development of LISP is based on the trade-off of pulling low volume parts ahead in the input sequence while delaying the high volume parts. We propose the use of the heuristic as a means to achieve both of the following: (a) to improve customer service levels in terms of due date performance measures given a fixed size for re-sequencing buffers; and (b) to reduce re-sequencing buffer sizes given target levels of customer performance.

Least In-Sequence Probability Heuristic for Mixed-Volume Production Lines

I. INTRODUCTION

The focus of this study is the development and subsequent analysis of an efficient control policy for deployment within hybrid flexible flow systems (FFSs) functioning in a build-to-order environment. Typically, within such a production system, the product demand is a sequence of mixed-volume part types. Given the hybrid nature of the system, although parts enter and exit in a sequence, within the system however, part types may follow different paths on parallel workstations. Further, in general, processing times are stochastic and, owing to the limited process capabilities of successive processing stations, parts might well be delayed due to quality concerns or machine failures. As a result, parts could violate prescribed due dates.

The above problem is often experienced in modern automotive plants. The competitive requirements to reduce delivery times, cut costs, and improve quality have spurred the development and implementation of innovative concepts such as In-Line Vehicle Sequencing (ILVS) [1][2][3] and Just-In-Sequence component delivery (JIS) [4] in automotive production. The ILVS philosophy, a sobriquet for lean production, integrates and establishes the stability for final assembly and sales for a certain period of time (typically a month), with segmentation fixed several days (typically 10) ahead, so that suppliers know exactly what is required [1]. The JIS approach notifies vendors of the exact sequence for delivering parts and materials with several days' notice or lead-time.

Final assembly sequence traditionally has received the most attention in automotive production since more than 60% of the assembly plant labor is concentrated in this area [2]. A good final assembly sequence is one wherein vehicles with high-content options are spaced apart from each other to balance the work content. Stability in final assembly allows one to move towards JIS delivery of components [4] further improving the efficiency of automotive production. For the success of JIS, it is crucial for component suppliers to have the required part types available on time, as per schedules dictated by the final assembly sequence. Many suppliers of components such as bumpers, seats, door panels, and especially frame production (which we view as a special case of an automotive component) to the final assembly line within automotive plants, are organized on an FFS basis.

From a scheduling viewpoint, Rachamadugu and Stecke [5] dichotomize FFSs into the following two types: flexible assembly systems (FASs) and flexible transfer lines (FTLs). Although work flows unidirectionally within each of these systems, whereas FTLs are able to simultaneously process a larger variety of part types, FASs are geared for higher production rates. Our focus in this paper is on the sequencing aspects of FASs, and more specifically, on a variant of the FAS class in which parallel paths exist for accommodating different families of parts produced (see Figure 1).

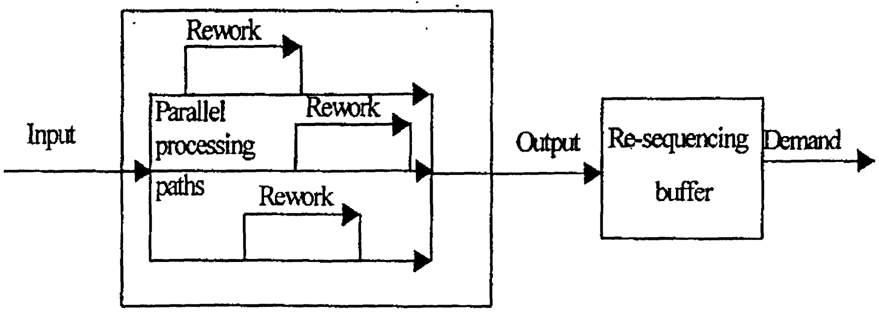


Figure 1. Hybrid FAS schematic

Further, within each part family line itself, rework stations run parallel to the main production lines to cater to parts that fail inspection at a given production stage. For instance, within the paint shop, some parts may get delayed because of the need for spot repair or may even require complete re-processing through a paint booth. Such delays due to quality concerns are common. Owing to these delays, parts often fall out-of-sequence and fail to meet due dates. To compensate for such manufacturing disturbances, component suppliers install buffers at the end of production lines in their attempt to re-sequence parts and match the original demand sequence (see Figure 2). For a more detailed exposition on sequencing issues in the automotive assembly context, refer to [2][3][4].

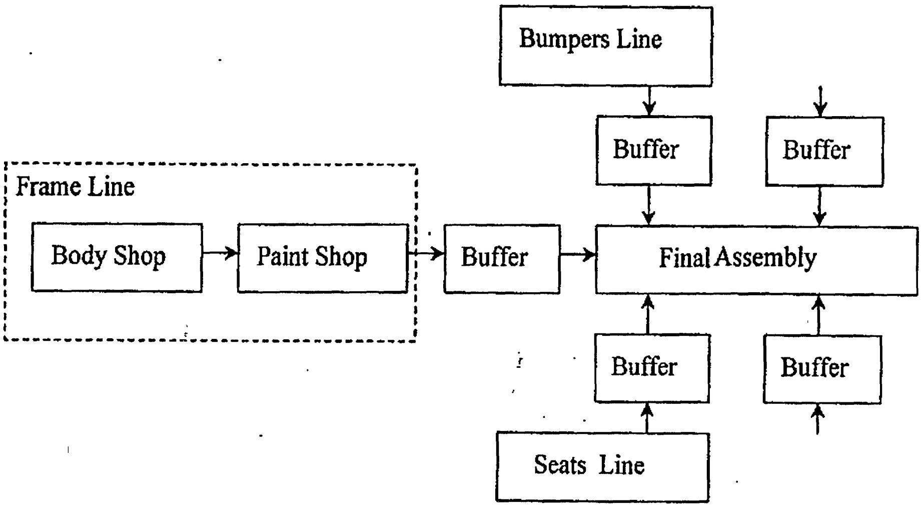


Figure 2. Typical automotive assembly line

II. PROBLEM DESCRIPTION

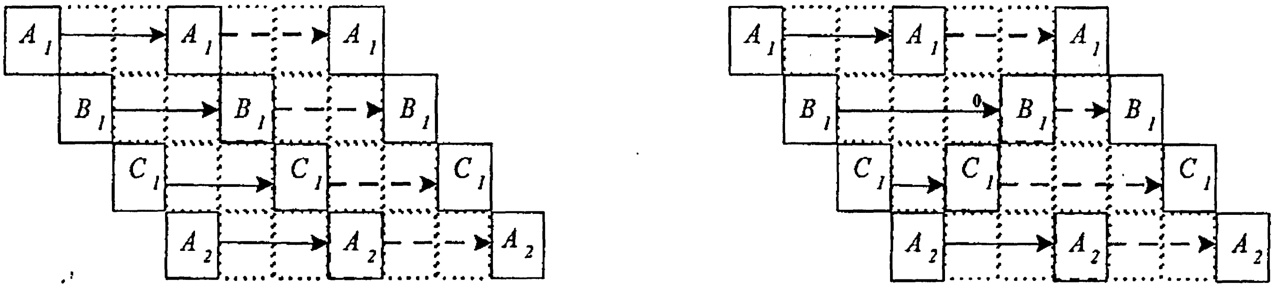
The most important feature of the FAS considered in this study is that the demand sequence of part types is known and fixed for a given period of time. Further, the different part types that constitute the demand sequence could have different frequencies of occurrence in a range specified from low to high. For example, in a demand sequence of 100 parts, with 80 parts of type A and 20 parts of type B, it is clear that the type A parts are the high frequency parts. Henceforth, part types with a low frequency of occurrence in the demand sequence are termed "low volume parts" while those with a high frequency are termed "high volume parts". Interestingly, for different sections of the demand sequence, the frequency of occurrence for a given part type is not fixed. For instance, in the above example, if we consider the first 20 parts in the sequence, it may happen that 15 of them turn out to be parts of type B. Clearly, for this section of the demand sequence, the type B parts are not the low volume parts. We exploit this property of the demand sequence in the development of the sequencing heuristic suggested in the paper.

After the specification of the demand sequence, one must decide the order in which the parts are to be processed through the production system. One possible approach would be to schedule parts according to their due dates, i.e., process the parts using the earliest due date sequencing heuristic. By scheduling production in this manner, the input sequence is in the same order as the demand sequence.

Because of the stochastic nature of the production system, some parts may spend more than the expected time in the system. As was pointed out in section 1, this additional processing time typically results from inefficient process capabilities of the processing stations. Since we do not assume any scrap from the production system, if a part fails a quality inspection, it gets routed to a rework station. Note that as the process capability of the production system decreases, the probability that a part requires rework increases. While being reworked, the part can be delayed in the system to a point in time that it is deemed late. As a consequence of this delay, the output sequence from the production system does not match the demand sequence. In an attempt to reconstruct the output sequence to resemble the demand sequence as closely as possible, we assume the presence of a re-sequencing buffer at the final stage of production. Finally, we develop a sequencing heuristic that alters the input sequence in an attempt to improve the probability of satisfying the demand sequence.

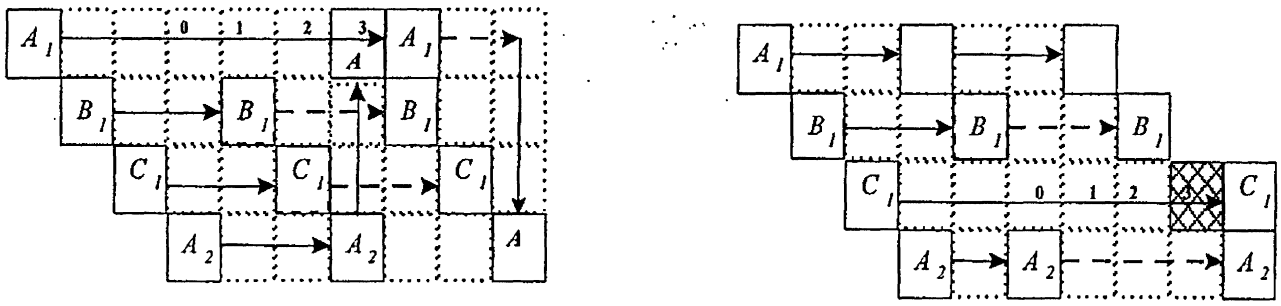
III. PROBLEM ANALYSIS

We use the re-sequencing buffer as a means of achieving the demand sequence to the extent possible. Because we allow parts to be temporarily stored in the buffer, a part is deemed late only when it is unavailable to occupy its expected position in the demand sequence. Based on this definition, we find it expedient to identify the following four scenarios for exposition purposes (see Figure 3):



(a) Scenario 1. Parts in sequence

(b) Scenario 2. Part on time, but out out of sequence



(c) Scenario 3. Part late, but substituted for

(d) Scenario 4. Part late and no substitution possible

Figure 3. Possible dispositions for parts flowing through an FAS

(1) All parts are processed through the system without delay, enter the buffer on time, and exit the system in accordance with the demand sequence meeting stipulated due dates (Figure 3 (a)).

(2) A part (depicted as B_1) experiences delay while processing, and although is out-of-sequence prior to entering the buffer, is still able to exit the system on time meeting the demand sequence (Figure 3 (b)).

(3) A part (depicted as A_1) experiences delay while processing and enters the buffer after violating its expected due date. From Figure 3 (c) (scenario 3), it is apparent that A_1 is three positions out-of-sequence. However, an identical type of part (depicted as A_2), having arrived into the buffer on time, substitutes for A_1, thereby meeting A_1's due date, while A_1 substitutes for A_2 which was scheduled to depart the system at a later due date. Thus, all parts exit the system in accordance with the demand sequence.

(4) A part (depicted as C_1) enters the buffer beyond its due date after experiencing delay in the system. Figure 3 (d) shows C_1 to be four positions out-of-sequence. Further, because part substitution is not possible, the demand sequence is violated.

So to minimize the number of parts that fall out of sequence and fail to meet due dates, a part substitution approach can be adopted. Using this approach, if a part is late and an identical part already resides in the buffer, then the "early part" can substitute for the "late part", thus alleviating the violation of due dates. However, an identical part may not always be available for substitution. Reasoning along these lines, it is clear that the high volume parts have a higher probability of meeting due dates than the low volume parts because, for the latter, the probability of two identical parts being present in the re-sequencing buffer simultaneously to effect substitution is lower. In fact, in many instances, a low volume part may be unique with no opportunity for substitution.

Based on the above discussion it seems intuitively clear that to increase the probability of the low volume parts meeting due dates, one can introduce them earlier than scheduled in the input production sequence. By scheduling low volume parts early, more time is allowed for contingencies (such as rework) while in process and the probability of maintaining the demand sequence, and thereby meeting due dates, increases. However, the early introduction of low volume parts results in delaying the high volume ones. The question that needs to be addressed therefore, is how to evaluate the trade-off between introducing low volume parts early and delaying the high volume ones. We suggest the ***least in-sequence probability*** (LISP) heuristic to address the above trade-off decision.

IV. THE LISP HEURISTIC ALGORITHM

To illustrate the idea behind LISP, consider the decision-making situation presented in Figure 4. The demand sequence consists of 10 parts of 4 different types (A, B, C, and D). One question is which part of what type should be input into production first. A straightforward way to select the next part to input is to apply a due date rule and select a part of type A. Note here that, from this point on, keeping in context with the focus of our paper, reference to a "due date" for a part is to be interpreted as the position in sequence for the part in the demand sequence. Accordingly, from Figure 4, the demand sequence shown (A B C D A C A B A) is in increasing order of "due dates". However, as has been shown in section III, in a stochastic production environment, high-volume part types have a higher probability of satisfying their demand positions than low-volume ones. Thus, to improve the probability of the output sequence of parts being as close to the demand sequence as possible, we choose the part that has the minimum probability of satisfying its due date. Importantly, the size of the re-sequencing buffer is a critical parameter that needs to be considered. Consider the case when the buffer size is 4. Assuming that all of the parts have been processed and are available for "re-sequencing" at time TNOW, Figure 4 shows that the first part of type *A* can meet its due date if it is not more than 4 positions out-of-sequence. Further, and importantly, since part B's due date is "one position" after part A, Figure 4 shows that at time TNOW, part B will have an additional slot for re-sequencing, which could alternatively be viewed as an added (virtual) re-sequencing buffer slot. Accordingly, part B will satisfy its due date if it is not more than (4 + 1 =) 5 positions out-of-sequence. Reasoning along these lines, the number of additional re-sequencing slots for part C is 6, while for part D it is 7.

We now need to calculate the probabilities of parts occupying appropriate positions in the demand sequence. For the above example, the probability of part A maintaining its position in the demand sequence (and thereby meeting its due date) is the sum of probabilities of it being in sequence. Let $P_i^{(j)}$ denote the probability of part 'j' being 'i' positions out-of-sequence, and P(j, x) denote the probability of part 'j' being not more than 'x' positions out-of-sequence (an outcome that ensures its correct placement in the demand sequence).

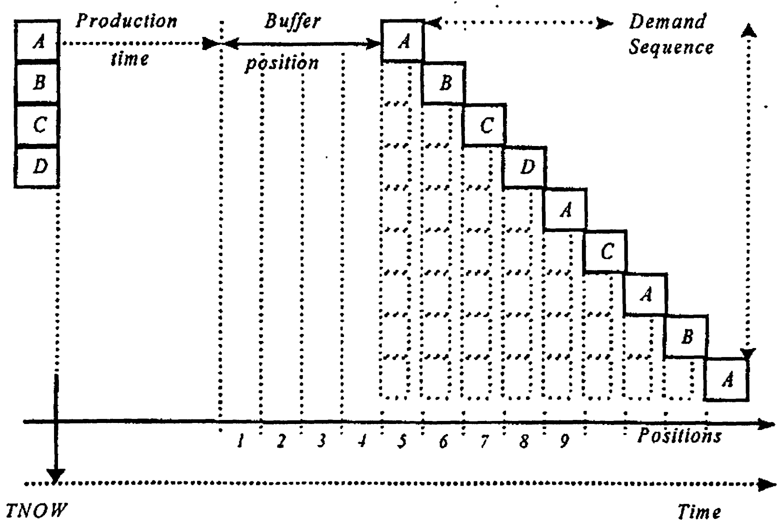


Figure 4. Re-sequencing buffer schematic

Then,

$$P(A, 4) = P_0^{(A)} + P_1^{(A)} + P_2^{(A)} + P_3^{(A)} + P_4^{(A)}$$

Similarly,

$$P(B, 5) = P_0^{(B)} + P_1^{(B)} + P_2^{(B)} + P_3^{(B)} + P_4^{(B)} + P_5^{(B)}$$

$$P(C, 6) = P_0^{(C)} + P_1^{(C)} + P_2^{(C)} + P_3^{(C)} + P_4^{(C)} + P_5^{(C)} + P_6^{(C)}$$

$$P(D, 7) = P_0^{(D)} + P_1^{(D)} + P_2^{(D)} + P_3^{(D)} + P_4^{(D)} + P_5^{(D)} + P_6^{(D)} + P_7^{(D)}$$

In general, the probability of part 'j' being not more than 'x' positions out-of-sequence is

$$P(j,x) = \sum_{i=0}^{x} P_i^{(j)} , \qquad (1)$$

where 'x' denotes the total number of (real and virtual) re-sequencing buffers available. P(j, x), therefore, can be interpreted to mean the probability that a part meets its due date (by maintaining its appropriate position in the demand sequence) given the possibility of re-sequencing within the (real and virtual) buffer slots.

The above probabilities (i.e., the $P_i^{(j)}$'s) can be calculated with ease if the probability distribution functions (PDFs) for each $P_i^{(j)}$ are known. A straightforward way to compute the required PDFs is through a simulation process. We now illustrate this procedure.

We use the same demand sequence shown in Figure 4 for purposes of simulation. Using this sequence as the input sequence into a simulation model for a hypothetical FAS, the following output sequence was observed: A C B A C A D A B. The simulation result for a single replication of the model is reproduced below in Table I. The first column (N) represents consecutive sequence numbers for parts, for both the input and output sequence. The next two columns represent the input sequence for each part along with sequence numbers ordered consecutively by part type, so that the first part A has an ordered sequence number (OSN) of 1, the second part A, an OSN of 2, and so on. The last two columns represent similar OSNs for the output sequence.

TABLE I
SIMULATION RESULT FOR THE ASSUMED DEMAND SEQUENCE

N	Input		Output	
	Part	OSN	Part	OSN
1	A	1	A	1
2	B	1	C	1
3	C	1	B	1
4	D	1	A	2
5	A	2	C	2
6	C	2	A	3
7	A	3	D	1
8	B	2	A	4
9	A	4	B	2

TABLE II
NUMBER OF POSITIONS OUT OF SEQUENCE FOR PARTS

Input		N_{input}	N_{output}	NPOS
Part	OSN			
A	1	1	1	0
B	1	2	3	1
C	1	3	2	0
D	1	4	7	3
A	2	5	4	0
C	2	6	5	0
A	3	7	6	0
B	2	8	9	1
A	4	9	8	0

We use Table I to construct Table II. The first two columns in Table II are reproduced from Table I (columns 2 and 3). The third column in Table II (N_{input}) represents the consecutive sequence position for parts in the input sequence and is essentially Table I's column 1. The fourth column in Table II (N_{output}) represents the sequence positions for parts in the output sequence, relative to their assigned sequence positions shown in N_{input}, resulting from the simulation experiment. Table II's last column represents the number of positions out-of-sequence (NPOS) for a given part.

NPOS is computed as follows. For each part ‘j’,

$$\text{NPOS} = 0, \text{ if } N_{\text{output}} \leq N_{\text{input}}\text{; and}$$

$$\text{NPOS} = N_{\text{output}} - N_{\text{input}}, \text{ if } N_{\text{output}} > N_{\text{input}}$$

Table III shows the number of occurrences for increasing values of NPOS for each part for 100 simulation replications. The first column of Table III represents parts along with their ordered sequence numbers, using the notation (j, OSN). For purposes of illustration, in Table III, we interpret the highlighted cell entry to imply that there were 5 occurrences in the 100 replications when part C (with OSN = 1) was found to be two positions out-of-sequence.

TABLE III
NUMBER OF OCCURRENCES FOR ‘NPOS’ FOR EACH PART

NPOS → (Part, OSN) ↓	0	1	2	3	4	5	6	7	8	9	Total
(A,1)	80	15	3	1	1	0	0	0	0	0	100
(B,1)	75	10	8	3	2	1	1	0	0	0	100
(C,1)	75	10	5	4	3	2	1	0	0	0	100
(D,1)	50	25	10	6	5	4	0	0	0	0	100
(A,2)	85	10	5	0	0	0	0	0	0	0	100
(C,2)	70	20	7	3	0	0	0	0	0	0	100
(A,3)	80	15	5	0	0	0	0	0	0	0	100
(B,2)	75	25	0	0	0	0	0	0	0	0	100
(A,4)	10	0	0	0	0	0	0	0	0	0	100

In Table IV, we compute $P_i^{(j)}$, the probability of the ‘j^th^’ part being ‘i’ positions out-of-sequence, by dividing the number of occurrences by the number of replications. Cell entries depict the $P_i^{(j)}$ values.

TABLE IV
COMPUTED VALUES FOR ‘$P_i^{(j)}$’ FOR EACH PART

NPOS → (Part, OSN) ↓	0	1	2	3	4	5	6	7	8	9	Total
(A,1)	0.8	0.1	0.0	0.01	0.01	0	0	0	0	0	1
(B,1)	0.7	0.1	0.0	0.03	0.02	0.0	0.0	0	0	0	1
(C,1)	0.7	0.1	0.0	0.04	0.03	0.0	0.0	0	0	0	1
(D,1)	0.5	0.2	0.1	0.06	0.05	0.0	0	0	0	0	1
(A,2)	0.8	0.1	0.0	0	0	0	0	0	0	0	1
(C,2)	0.7	0.2	0.0	0.03	0	0	0	0	0	0	1
(A,3)	0.8	0.1	0.0	0	0	0	0	0	0	0	1
(B,2)	0.7	0.2	0	0	0	0	0	0	0	0	1
(A,4)	1	0	0	0	0	0	0	0	0	0	1

Finally in Table V, we compute P(j, x) (refer to equation 1) for each NPOS value. Recall that P(j, x) is the probability of part 'j' being not more than 'x' positions out-of-sequence and is interpreted as the probability that a part meets its due date (by maintaining its appropriate position in the demand sequence) given the possibility of re-sequencing within the (real and virtual) buffer slots. For the highlighted cell then, the probability of part C (with OSN = 1) being not more than 2 positions out-of-sequence is 0.9.

TABLE V
COMPUTED VALUES FOR P(j, x) FOR EACH PART

NPOS → (Part, OSN) ↓	0	1	2	3	4	5	6	7	8	9
(A,1)	0.8	0.95	0.98	0.99	1	1	1	1	1	1
(B,1)	0.7	0.85	0.93	0.96	0.98	0.99	1	1	1	1
(C,1)	0.7	0.85	0.9	0.94	0.97	0.99	1	1	1	1
(D,1)	0.5	0.75	0.85	0.91	0.96	1	1	1	1	1
(A,2)	0.8	0.95	1	1	1	1	1	1	1	1
(C,2)	0.7	0.9	0.97	1	1	1	1	1	1	1
(A,3)	0.8	0.95	1	1	1	1	1	1	1	1
(B,2)	0.7	1	1	1	1	1	1	1	1	1
(A,4)	1	1	1	1	1	1	1	1	1	1

We now describe the LISP heuristic algorithm using the notation shown in Table VI.

TABLE VI
NOTATIONS FOR THE HEURISTIC SEQUENCING ALGORITHM

B	size of the re-sequencing buffer (i.e., number of re-sequencing slots available in the buffer)
D	the set of parts in the demand sequence
D_j	j^{th} part in the demand sequence (D) set
J	sequence number of a part in the set D
x_j	the maximum allowable number of positions out-of-sequence for the j^{th} part
$P(D_j, x_j)$	the probability that part D_j is not more than x positions out of sequence
S	the set of parts in the input sequence
S_i	i^{th} part in the set S
I	sequence number of a part in the set S

The steps for the LISP algorithm are now given.

The LISP Algorithm

Step 1. Set S = {∅} and I = 1.
Step 2. For all $j \in D$, compute $x_j = J - I + B$.
Step 3. Select D_k, such that there is no $r \neq k$ for which $P(D_r, x_r) < P(D_k, x_k)$, and there is no $r < k$ for which $P(D_r, x_r) = P(D_k, X_k)$.
Step 4. Set $S_i = D_k$. Remove D_k from D.
Step 5. If D = {∅} then stop. Else, I = I + 1. Go to step 2.

We illustrate the algorithm using the data generated in Table V for the example considered. For exposition, we assume a buffer size B = 2. Stepping through the algorithm, the resulting output for successive iterations is summarized in Table VII. Cell values in Table VII represent the $P(D_j, x_j)$ probabilities for the D_j^{th} part being not more than x (which is equal to (J – I + B)) buffer positions out-of-sequence. The $P(D_j, x_j)$ probabilities are obtained from Table V.

TABLE VII
INPUT SEQUENCE GENERATION USING THE SEQUENCING HEURISTIC

(Part, OSN)	(A,1)	(B,1)	(C,1)	(D,1)	(A,2)	(C,2)	(A,3)	(B,2)	(A,4)	New Input equence
J	1	2	3	4	5	6	7	8	9	
1	0.98	0.96	0.97	1	1	1	1	1	1	(B, 1)
2	0.95	-	0.94	0.96	1	1	1	1	1	(C, 1)
3	[illegible]	-	-	0.91	1	1	1	1	1	(A, 1)
4	-	-	-	0.85	1	1	1	1	1	(D, 1)
5	-	-	-	-	[illegible]	1	1	1	1	(A, 2)
6	-	-	-	-	-	[illegible]	1	1	1	(C, 2)
7	-	-	-	-	-	-	[illegible]	1	1	(A, 3)
8	-	-	-	-	-	-	-	[illegible]	1	(B, 2)
9	-	-	-	-	-	-	-	-	1	(A, 4)

For example, for I =1, part (A,1) will satisfy its due date requirement if it is not more than 2 positions out-of- sequence, part (B,1) - not more than 3 positions out-of-sequence, part (C,1) - not more than 4 positions out of sequence, and so on. From Table V, looking down the NPOS row, probability P((A,1), 2) = 0.98, probability P((B, 1), 3) = 0.96, probability P((C, 1), 4) = 0.97, probability P((D, 1), 5) = 1, and so on. These probability values appear in Table VII along the row where I = 1. Since part (B, 1) has the minimum probability (= 0.96) of satisfying its due date, it is chosen as the first part in the input sequence. With one buffer slot reserved for (B, 1), a single slot now remains in the re-sequencing buffer. Therefore, part (A,1) will now satisfy its due date requirement if it is not more than 1 position out-of- sequence, part (C,1) - not more than 3 positions out of sequence, part (D,1) - not more than 4 positions out of sequence, and so on. Likewise, (C, 1) constitutes the second part in the input sequence because it has the smallest $P(D_j, x_j)$ probability (= 0.94) from among the remaining parts for I = 2. Reasoning along these lines, we determine the new input sequence as shown in Table VII. The new input sequence is seen to be: B C A D A C A B A. Note here that for I = 5, 7, 8, and 9, the $P(D_j, x_j)$ probabilities are identical and equal to 1. In such a situation, we select the part with minimal sequence number J. We now study the performance of the suggested heuristic through simulation experiments on a hypothetical model of an FAS.

V. SIMULATION DETAILS AND RESULTS

For purposes of simulation, we considered an FAS consisting of a single processing station, a rework station, and an output re-sequencing buffer. The ARENA 3.0 simulation package (Systems Modeling Corp.) was used, into which Visual Basic routines were linked to capture the more complex logic. For the assumed system, a demand sequence of 100 parts comprising of 4 different part types (A, B, C, and D) was considered. Part processing times for both primary and rework operations were exponentially distributed. Without loss of generality, the distribution parameter for each part type was assumed identical. Further, to enhance the possibility of out-of-sequence situations, we resorted to the following: (a) the probability of failing inspection at the primary processing station was intentionally chosen to be a high value of 0.4; (b) the distribution parameter for the rework operation time was also intentionally chosen much larger (50 minutes) than the primary processing operation time (10 minutes).

Three possible part mix combinations together with different re-sequencing buffer sizes were considered. Details are shown in Table VIII. Part mix entries in Table VIII are in percentage values.

For all cases analyzed, the number of parts late is notably lower for LISP than for the EDD rule. Notice further that the NPOS values using the EDD rule are considerably less than the corresponding values for LISP. However, this apparent superiority of the EDD rule is indeed misleading. Importantly, we note here that the NPOS values are part specific, and that the EDD rule does not distinguish between the low-volume and high-volume parts. Therefore, although the NPOS values using the EDD rule are better, the number of parts late are considerably worse than those for LISP. By pulling ahead the low-volume parts, the suggested heuristic ensures that they meet stipulated due dates even in the presence of unexpected delays within the system. With the high-volume parts however, although larger NPOS values result, part substitution ensures that due dates are met.

TABLE VIII
SIMULATION RESULTS FOR TEST CASES CONSIDERED

Case Num.	Part Mix				Buffer size	EDD				LISP			
	A (%)	B (%)	C (%)	D (%)		NPOS	%	Parts late	%	NPOS	%	Parts late	%
1	60	20	15	5	15	74241	37.12	2693	1.35	108506	54.25	476	0.24
2	60	20	15	5	20	74241	37.12	1096	0.55	101273	50.64	229	0.11
3	60	20	15	5	25	74241	37.12	345	0.17	90671	45.34	90	0.05
4	60	20	15	5	30	74241	37.12	138	0.07	81595	40.80	52	0.03
5	60	20	15	5	35	74241	37.12	38	0.02	76307	38.15	25	0.01
6	50	25	15	10	15	71102	35.55	3732	1.87	97702	48.85	1012	0.51
7	50	25	15	10	20	71102	35.55	1486	0.74	93783	46.89	284	0.14
8	50	25	15	10	25	71102	35.55	615	0.31	83495	41.75	125	0.06
9	50	25	15	10	30	71102	35.55	232	0.12	76909	38.45	45	0.02
10	50	25	15	10	35	71102	35.55	60	0.03	73858	36.93	25	0.01
11	40	30	20	10	15	82918	41.46	3122	1.56	99971	49.99	1547	0.77
12	40	30	20	10	20	82918	41.46	1334	0.67	94649	47.32	364	0.18
13	40	30	20	10	25	82918	41.46	582	0.29	89610	44.81	104	0.05
14	40	30	20	10	30	82918	41.46	229	0.11	86369	43.18	44	0.02
15	40	30	20	10	35	82918	41.46	60	0.03	84594	42.30	25	0.01

In this example the range of part mix combinations reflects the distribution of parts so that the high-volume and low-volume parts are easily identified for cases 1 and 2. However, for cases 5 and 6, the distinction is not as obvious. The results are commensurate with the part mix distributions. For cases 1 and 2, with part type A clearly the high-volume type, LISP gives significant improvement over EDD in terms of the number of late parts. As the product mix becomes more balanced, the improvement in results reduces.

Figure 5 graphs the percentage of parts late against increasing buffer sizes for the 3 different part-mix combinations assumed (see table VIII). Figures 5 (a), (b), and (c) clearly show that for a given buffer size, LISP exhibits superior performance in terms of the percentage of parts late. Alternatively viewed, for a specified customer service level (percentage of late parts), LISP would result in smaller buffer requirements.

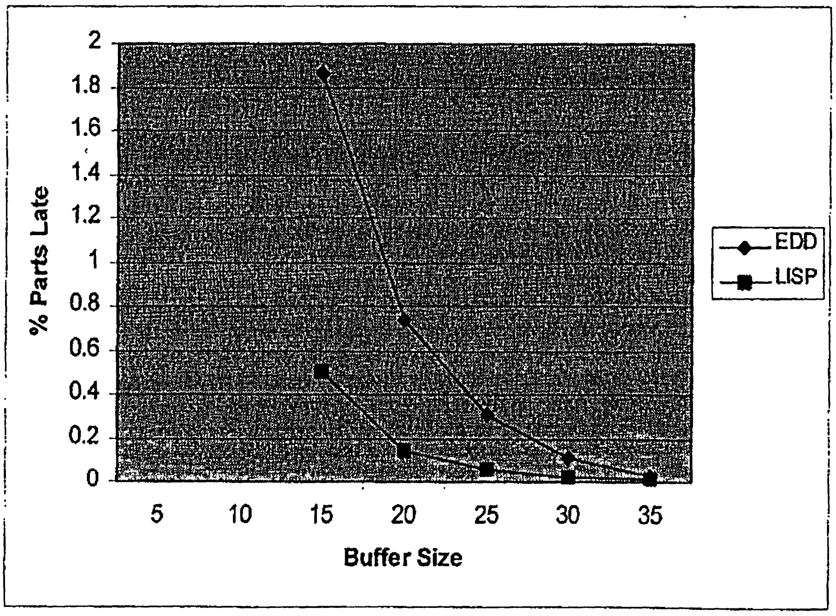


Figure 5(a): EDD vs. LISP performance for part mix (A – 60%, B – 20%, C – 15%, D – 5%)

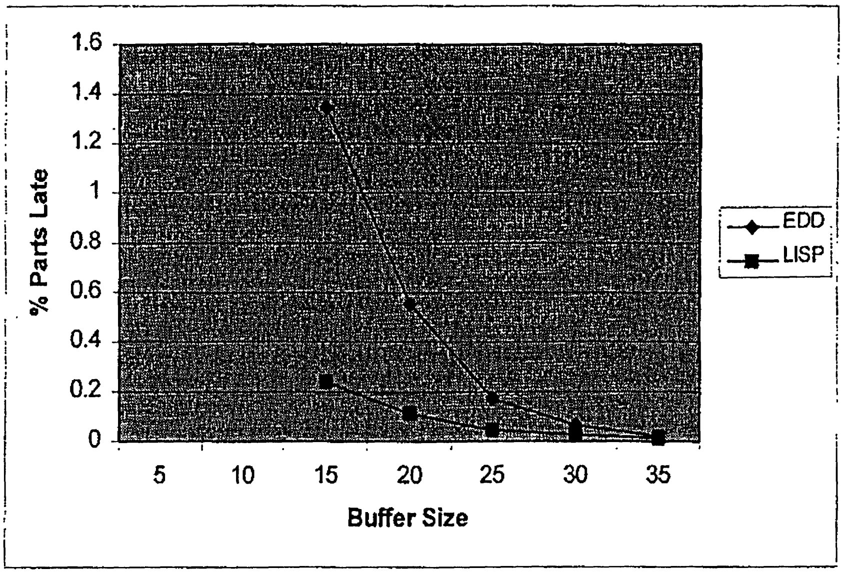


Figure 5(b): EDD vs. LISP performance for part mix (A – 50%, B – 25%, C – 15%, D – 10%)

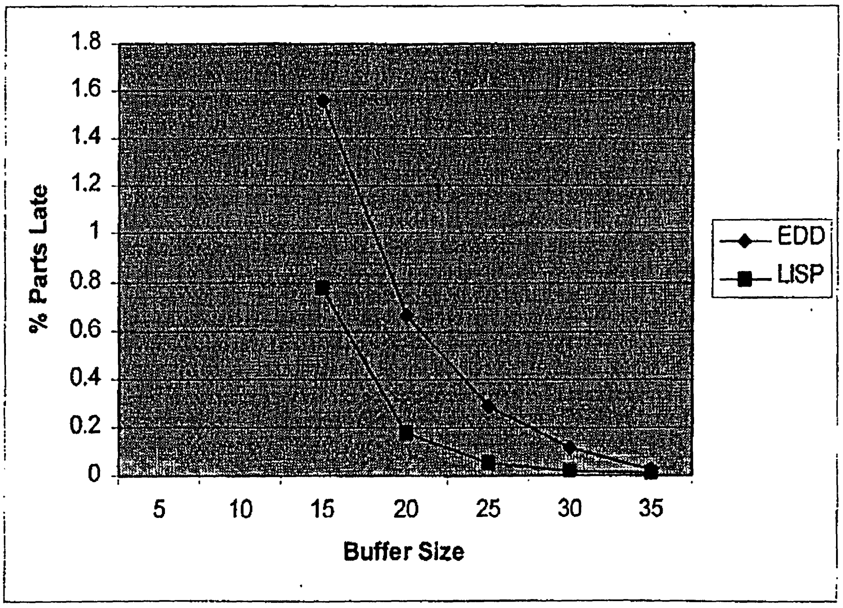


Figure 5(c): EDD vs. LISP performance for part mix (A – 40%, B – 20%, C – 30%, D – 10%)

The graphs shown in Figure 5 also highlight another important point with significant practical ramifications. In some cases, to maintain a given customer service level (i.e., parts delivered on time) the production system might require relatively large re-sequencing buffers. According to Sawer [3], Wixom – Ford's lead plant for the company's ILVS in North America - has a large Automatic Storage and Retrieval System (AS/RS) on-site that holds no less than 250 vehicles, or about five hours worth of production. In the context of buffer reduction, the simulation results of our study highlight the scope for potential improvement. Specifically, Figure 5 (a) shows that for a 99.8% customer service level, LISP causes a reduction in buffer size by approximately 30% when compared with the EDD rule. Given this reduction in buffer size, the corresponding AS/RS size required for ILVS implementation could be thereby be reduced, with significant dollar savings.

To provide insight into the behavior of the suggested heuristic, for the hypothetical FAS assumed, the probability distribution for increasing NPOS values for two different cases are plotted in Figure 6. For the first case, the distribution for NPOS values obtained for two (arbitrarily chosen) individual parts (with sequence numbers 63 and 65), from a demand sequence in which each part was assumed unique, is shown in Figure 6 (a). Further, the distribution for the NPOS values for the same two sequenced numbered parts but now as constituent members (of types A and D) from the product mix shown as case 1 in Table VIII, is plotted in Figure 6 (b). It is obvious from these graphs that the probabilities for the parts to satisfy their due dates increases as the volume of parts in a product mix increases. In Figure 6(b), the probability for the high volume part type A to meet its due date is notably higher than that for the low volume part type D. However, with each individual part considered unique, the probabilities for meeting due dates are the same as seen in Figure 6(a). The differences in probability distributions for the NPOS values for different parts are caused entirely by the volume of the different part types in the demand sequence.

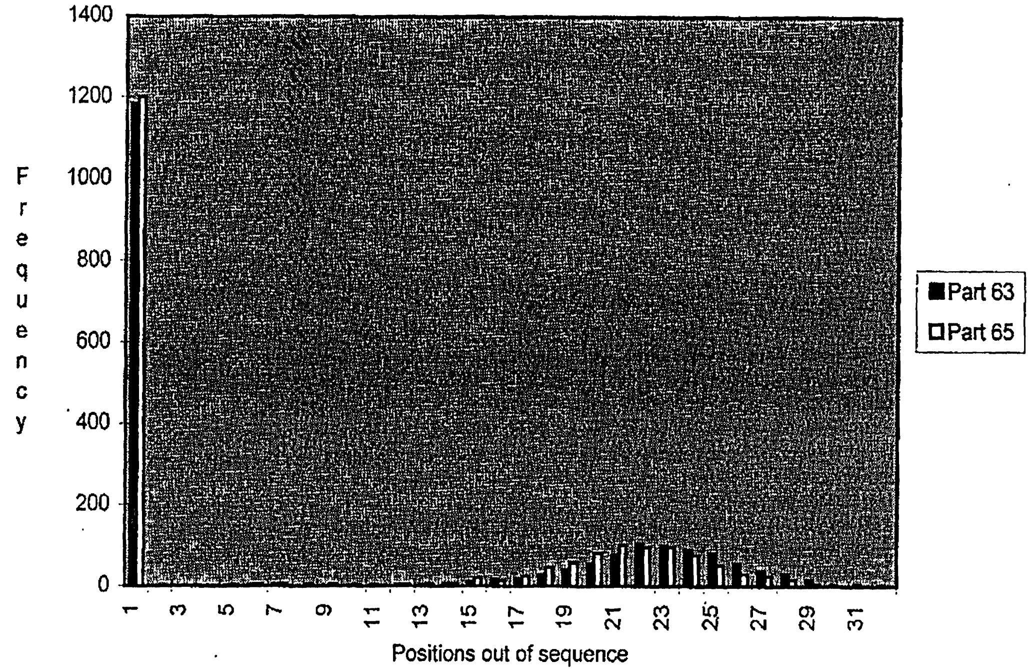


Figure 6(a). Probability distribution of NPOS for individual parts

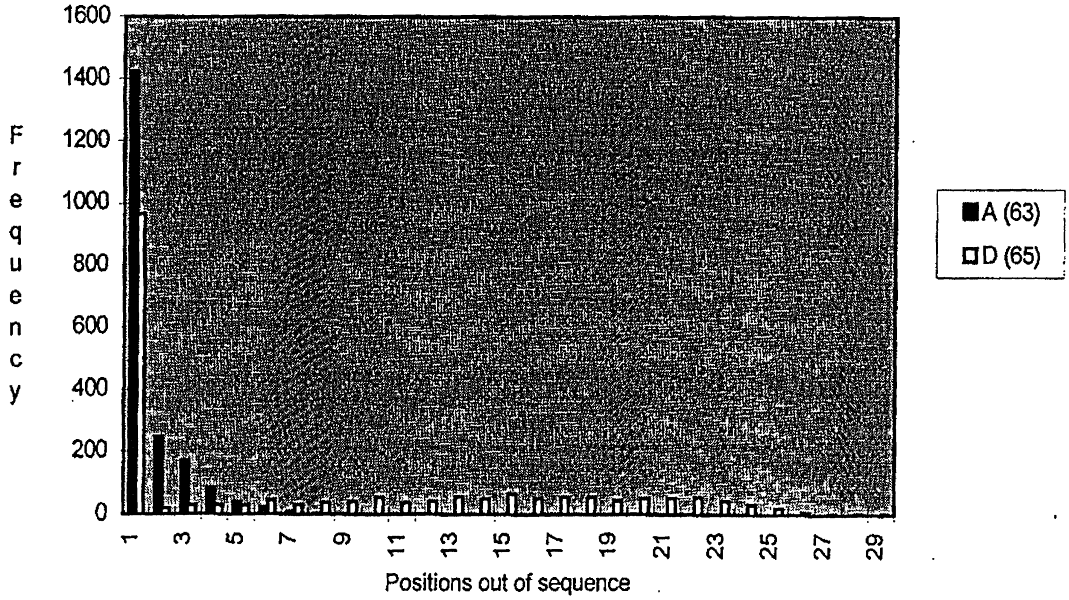


Figure 6 (b). Probability distribution of NPOS for different part types

VI. CONCLUSIONS

In this paper, we have proposed and analyzed an efficient control policy for deployment within flexible assembly systems functioning in a build-to-order environment. Typically, within such a production system, product demand is a sequence of mixed-volume part types. Further, the presence of stochastic processing times and the limited process capabilities of work stations, results in parts being delayed due to quality concerns or machine failures. As a result, parts could well violate prescribed due dates and demand sequences.

Simulation results highlight the efficacy of the suggested heuristic in meeting demand requirements for a hypothetical study system. Specifically, by first choosing parts with low probabilities of meeting due dates in the input sequence, the heuristic compensates for delays due to random system disturbances, by virtue of which stipulated due dates are met.

We propose the use of the heuristic as a means to achieve both of the following: (a) to improve customer service levels in terms of due date performance measures given a fixed size for re-sequencing buffers; and (b) to reduce re-sequencing buffer sizes given target levels of customer performance.

Finally, we propose the architecture shown in Figure 7 as a means for real-time FAS control. The FAS control system accumulates historical system performance data. This historical data is used to compute and set statistical parameters for simulation modeling purposes. The control system is also used to effect simulation system initialization. Along with the above, and given a prescribed demand sequence, the simulation model computes probabilities for different parts occupying appropriate positions in the demand sequence. The heuristic algorithm uses these probabilities to construct the new input sequence, which in turn is supplied back to the FAS control system.

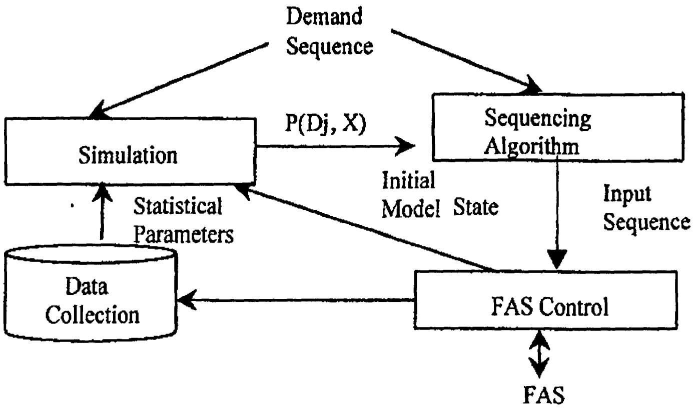


Figure 7. Real-time FAS control

REFERENCES

[1] Birch S., "The art of taking time out," *Automotive Manufacturing*, pp.16-19, April 1996.

[2] Morrison K. R., "Animation - A new dimension in computer simulation of automotive assembly processes," *Computers and Industrial Engineering*, vol. 21, pp. 547-551, 1991.

[3] Sawyer, C.A., "In-Line Sequencing: How it's done at Wixom", *Automotive Industries*, p.41, November 1994.

[4] Coffman, P.E. and Rachamadugu, R., "Impact of assembly sequencing on final assembly and component delivery,".SAE Report # SAE-P-338; 1998.

[5] Rachamadugu, R. and Stecke, K.E., "Classification and review of FMS scheduling procedures," *Production Planning and Control*, vol. 5, no.1, pp. 2-20, 1994.