A KNOWLEDGE BASED SYSTEM FOR BUFFER ALLOCATION IN UNRELIABLE PRODUCTION LINES

G. A. Vouros[1] and H. T. Papadopoulos[2]

Department of Mathematics, University of the Aegean, GR-832 00 Karlovassi, Samos, Greece

Scope and Purpose – This article describes ASBA2. This is a knowledge based system that solves the buffer allocation problem for unreliable production lines, with computational efficiency and very satisfactory results. The system aims to advice manufacturing system designers on allocating buffer space in unreliable (balanced or unbalanced) production lines.

[1]**George A. Vouros** is a Visiting Professor in Artificial Intelligence and Databases at the Department of Mathematics, University of the Aegean, Greece. He received his Ph.D. from the University of Athens in 1992. He has participated in many E.U. funded projects and he is the author of more than 20 papers concerning natural language engineering, conceptual modeling, and expert systems. He is member of AAAI, IEEE and member of the board of directors of the Hellenic Society for Artificial Intelligence.

[2]**H. T. Papadopoulos (Chrissoleon)** is an Assistant Professor of Operations Research in the Department of Mathematics at the University of the Aegean, Samos island, Greece. He received his B.Sc. (in Mathematics) from the Aristotle University of Thessaloniki, Greece, his M.Sc. (in Operations Research and Informatics) from the National Kapodistrian University of Athens and his Ph.D. in Operations Research and Industrial Engineering from the Department of Industrial Engineering of the University College, Ireland in 1989. Before joining the Department of Mathematics at the University of the Aegean, he was the Logistics and Administration Manager and Customer Services Sales Manager at Digital Equipment Corporation (DEC) Hellas. His research interests include stochastic modelling, design and analysis of manufacturing systems, production and operations management, optimization of queueing systems, Logistics and Purchasing management and development of Decision Support Systems. He is co-author of a book on the analysis and design of manufacturing systems, published by Chapman and Hall. He is a member of INFORMS (ORSA/TIMS).

Abstract - Optimisation of production lines' performance is a problem of great complexity, and therefore, of significant research interest. The problem may involve the optimisation of many conflicting objectives, such as increasing throughput and reducing work-in-process time. The majority of existing studies have used various heuristics and search methods based on operations research. These methods have been proved to be computationally inefficient, especially for large production lines. This paper presents ASBA2, a knowledge based system that determines near optimal buffer allocation plans, with the objective to maximise production lines' throughput. The allocation plan is calculated subject to a given amount of total buffer slots, in a computationally efficient way. ASBA2 operates in close cooperation with a simulation method, which provides ASBA2 with performance measures concerning production lines' behaviour. Moreover, to evaluate results provided by ASBA2, we have utilised an exact numerical algorithm for calculating the throughput of unreliable production lines.

1 Introduction and literature review

There is a wide range of modelling techniques that can be used to address manufacturing system design, operational and optimization problems. From the user's point of view it is important to differentiate between generative models and evaluative models (see [1] and Chapter 1, section 5 of Papadopoulos *et al.* [2]). Evaluative models are concerned with the evaluation of certain measures of performance, such as the throughput, the mean flow time and the mean queue lengths formed in front of the various workstations of the production line. On the other hand, generative models deal with optimization problems concerned with the scheduling of resources in a manufacturing system. Buffer allocation is a major optimisation problem faced by manufacturing systems designers as well as by researchers. It has to do with devising an allocation plan for distributing a certain amount of buffer space among

Figure 1: Evaluative model combined with a generative model

the intermediate buffers of a production line.

Evaluative and generative models can be combined by 'closing the loop' between them. A generative model can use feedback from an evaluative model to form and modify its decisions, as well as to check whether certain actions have achieved desired effects. Figure 1 illustrates such a configuration between an evaluative and a generative model. The evaluative model, in general, is used to obtain values of objective functions for a set of inputs. Such values describe production line's behaviour and are communicated to the generative model. The latter uses the obtained values in its search for an optimal solution.

For a systematic review of the existing literature in the area of evaluative and generative models of manufacturing systems, the interested reader is addressed, respectively, to a review paper by Papadopoulos and Heavey [4] and to the book by Papadopoulos *et al.* [2], Chapter 7, among others.

Although several researchers have studied the problem of optimizing buffers' allocation to maximize the efficiency of an unreliable production line, there is no method that can handle this problem for large production lines, in a computationally efficient way. Papadopoulos and Vidalis [10,11,12] have thoroughly investigated the buffer allocation problem in various types of production lines (balanced, semi-balanced reliable and unreliable lines). In their study, they have combined an efficient numerical algorithm (developed by Heavey, Papadopoulos and Browne [13]) for calculating the throughput of a production line and a very accurate search method for determining the optimal buffer capacities that maximize the throughput of the lines. However,

their method cannot solve the problem for production lines with more than twelve stations and intermediate buffers with more than 4 slots. The same difficulty is encountered in the methods proposed by Hillier and So [6], and Hillier, So and Boling [7]. These methods are based on comprehensive studies to characterize the optimal buffer allocation pattern. Authors have provided extensive numerical results for balanced lines with up to 6 stations and limited results for lines with up to 9 stations. Other relevant studies are those by Conway *et al.* [5] who conducted simulation experiments to study the buffer allocation problem, Powell [8] who studied the buffer allocation problem for *unbalanced* production lines and So [9] who dealt with the buffer allocation problem in a production line, having as objective to minimize the average work-in-process, provided a minimum required throughput is attained.

The objective of this paper is to present ASBA2, a knowledge based system for solving the buffer allocation problem in unreliable, balanced and unbalanced, production lines with computational efficiency. ASBA2 is an extension of a previous system, called Advisor System for Buffer Allocation (ASBA) which was developed for solving this problem for reliable production lines, i.e., lines with machines that were not allowed to fail ([3]). ASBA2 is a generative model that works in close cooperation with an evaluative - simulation model.

This paper is organized as follows. Section 2 states the problem and assumptions of the model, whereas, section 3 describes the proposed knowledge based system. In section 4, we provide numerical results obtained from this system. Finally, section 5 concludes the paper and some future research directions are suggested. The code listing of the proposed system is not given due to space reasons but it is available from the authors, upon request.

2 The model and the buffer allocation problem

In asynchronous production lines, when a workstasion has completed its processing and the next buffer has available space, then the processed part is passed on. Then, the workstation starts processing a new part that is taken from its buffer. In case

the buffer has no parts, the workstation remains idle until a new part is placed in the buffer. This type of line is subject to manufacturing blocking (or blocking after service, as it is known in the literature) and starving.

Assumptions of the model: It is assumed that the first station is never starved and the last station is never blocked. The processing (service) times at each workstation are assumed to be independent random variables following either the exponential or Erlang-k ($k \geq 2$) distribution, with mean service rates, μ_i, $i = 1, 2, \ldots, K$. The workstations of the line are subject to random failures according to a Poisson distribution. That is, the times to failures are assumed to follow the exponential distribution with mean failure rates β_i, $i = 1, 2, \ldots, K$. The repair times are also assumed to follow the exponential or Erlang-m distribution ($m \geq 1$), with mean repair rates r_i, $i = 1, 2, \ldots, K$.

Although the assumption concerned with the distribution of processing times is not so realistic, it has been used by many researchers as it facilitates modelling a production line by the well-known Markovian state method. According to this method, the line is modelled as a series or tandem queueing network with K single-server stations that correspond to the K workstations of the production line. This allows the manufacturing system designer to get insight into the various performance measures of the production line. At the preliminary design stage, it is very helpful to get a good initial idea of how the production line performs under various assumptions and then try to optimize its performance by fine-tuning the values of the decision variables. At the operational level, simulation and other techniques have to be utilized to cope with the on-line decisions that must be taken.

The K-station line has $K-1$ intermediate locations for buffers, labelled $B_2, B_3, \ldots, B_K$, in Figure 2.

The basic performance measures in the analysis of production lines are the throughput (or mean production rate) and the average work-in-process (WIP) or equivalently the average production (sojourn) time. So [9], and Papadopoulos and Vidalis [14] have been dealt with the optimal buffer allocation problem of production lines, having as target the minimization of the average WIP. WIP plays an important role

Figure 2: A K-station production line with $K-1$ intermediate buffers

carrying a significant cost for the whole production line.

The object of the current paper is the buffering of asynchronous, unreliable production lines with the assumptions given above. The objective is the maximization of lines' throughput, subject to a given total buffer space.

The optimal buffer allocation (OBA) problem: In mathematical terms, our problem could be stated as follows:

(P1) Find $B = (B_2, B_3, \ldots, B_K)$ so as to

$\max X_K(B)$

subject to:

$\sum_{i=2}^{K} B_i = N,$

$B_i \geq 0,$

B_i integer $(i = 2, 3, \ldots, K),$

where: N is a fixed nonnegative integer, denoting the total buffer space available in the production line.

$B = (B_2, B_3, \ldots, B_K)$ is the 'buffer vector', i.e., a vector with elements the buffer capacities of the $K-1$ buffers.

X_K, denotes the throughput of the K-station line. This is a function of the mean service rates of the K stations, μ_i, $(i = 1, 2, \ldots, K)$, of the coefficients of variation, CV_i, of the service times and the buffer capacities, B_i. It is also a function of higher moments of the service time distribution, but these are usually ignored, for simplicity. Besides, it has been experimentally, by simulation, proved that these higher moments

have a minor effect on the throughput of the production lines.

Methodology of investigation: To solve the optimal buffer allocation problem (P1), we have performed the following steps:

- (S1) We modeled the queueing process of the production line as a finite state, continuous time Markov chain, due to the assumption of the Erlang-k ($k \geq 1$) distribution for the processing times;

- (S2) To calculate the exact throughput of a production line we have utilised a very efficient algorithm developed by Heavey, Papadopoulos and Browne [13]. The algorithm solves the sparse system of the steady-state probabilities of the resulting Markov chain. The algorithm gives the exact throughput for any K-station line with finite intermediate buffers and phase-type service times. The number of states and the number of feasible allocations of N buffer slots among the $K-1$ intermediate buffer locations increase greatly with N and K. The latter is given by the formula:

$$\frac{(N+1)(N+2)\cdots(N+K)}{(K-2)!} \tag{1}$$

- (S2') As an alternative method to the above mentioned exact numerical methods, we developed a simulation algorithm that calculates the throughput of the production line as well as the idle, blocking and the mean waiting time in each workstation.

- (S3) To find the optimal buffer allocation (OBA) that maximizes the throughput of the line, we utilized the knowledge based system ASBA2, that has been developed specifically for solving this problem.

3 The knowledge based system

Allocation of buffers in a production line with a specific number of stations, is a complicated, critical task, that requires experience. The problem becomes much

more complicated when more realistic assumptions are incorporated into the system, i.e., when machines in the production line are unreliable. As already explained, this means that machines can break any time and need some time in order to be repaired. As already has been stated elsewhere [3], complexity, criticality, and the experience intensity of allocating buffers in a production line justify the need for an advisory system. In [3] we have investigated the use of a specific framework for representation and reasoning, according to which we have structured and encoded knowledge for solving the following problem: Given a total buffer size and a minimum required throughput, find a buffer allocation configuration that minimizes the average work-in-process subject to the minimum required throughput for a production line with reliable machines. The knowledge based system proposed, called ASBA, had been designed so as to enable the incorporation of constraints concerning specific peculiarities of production lines working in a specific context. The system proved to be efficient for achieving a near optimum solution to the stated problem. However, it deals only with reliable machines. This is considered to be a critical relaxation. The objective is to extend ASBA to deal effectively with the most general case, i.e., with unreliable unbalanced and balanced, production lines. As it has been stated previously, no computationally efficient technique exists for the solution of this problem, especially for large lines.

This section describes our investigation of applying A.I. techniques to solve the problem of devising buffer allocation plans for production lines with unreliable machines. The extended ASBA system (ASBA2) allocates space to buffers by taking into account restrictions concerning throughput and the total buffer size. The system does not consider work-in-progress, which is our next aim. Specifically, ASBA2 solves the buffer allocation problem stated in section 2: it calculates a near optimal buffer allocation that maximizes throughput subject to the constraint on the total buffer space.

To solve this problem, ASBA2 performs the following actions:

1. Allocates specific buffer size to each station;

2. Transfers buffer units from source to target stations according to specific criteria realised by heuristic rules;

3. Frees some buffer size when it is no needed or not in use at all;

4. Submits requests to a simulation model, to calculate the throughput of the production line, as well as the idle, blocking and average waiting time in each station, for a given buffer allocation plan.

This section presents the types of knowledge encoded in ASBA2 according to the specific framework developed for ASBA. These are the strategic, the operational, the control and the heuristic knowledge. Special emphasis will be given to operational and heuristic knowledge since it is this kind of knowledge that enables solving the problem in a computationally efficient way.

1. *Strategic knowledge:* This is knowledge about generic procedures followed during problem solving and about forming specific strategic decisions in problematic situations that arise. As already been described in [3], such knowledge is used to decide which goals to pursue, which constraints to relax, whether certain progress is made in achieving a specific goal, and to resolve trade-offs between requirements and constraints. Since ASBA2 deals only with maximizing throughput, it does not deal with trade-offs between constraints and requirements. Goals that the system pursues during reasoning include reducing the blocking time of a machine, allocating the total buffer size, and increasing throughput. The only constraint that can be relaxed is reducing the blocking time of any machine. However, ASBA2 may give up increasing the throughput of the production line when it can not achieve any progress with respect to that goal. A crucial, let us say, part of strategic knowledge concerns monitoring whether the system has achieved any progress towards increasing throughput. For instance, in achieving the goal: "reduce the blocking time of the 2nd station", the system performs a concrete action. To decide whether that action is successful, the system must monitor that the throughput of the production line has been increased.

Strategic knowledge in ASBA2 is implemented by PROLOG rules. These rules follow the generic structure of <*Preconditions, Strategy Body, Post Conditions*>.

Preconditions check the state of the ASBA2 for strategy execution. This state comprises the *physical configuration* of the production line and the ASBA2 *internal state*. The physical configuration is the current buffer allocation plan. ASBA2 internal state comprises the target goal of the system and the current throughput of the production line for the corresponding physical configuration. ASBA2 state is recorded in a 'decision frame'. A history of decision frames is kept with respect to the beginning of the ASBA2 operation for a particular problem. Decision frames guide and justify the execution of particular operations.

The *Body of a Strategy* represents a sequence of strategic steps. A strategic step can be a strategy or a specific operation.

Post Conditions are procedures that check whether the execution of the strategy achieved the target goal described in the ASBA2 internal state. Of particular interest here is the "Progress" procedure that helps monitoring progress towards a goal state.

Strategies, during execution, are instantiated to the specific circumstances to which they apply. Strategy instantiation results to a sequence of concrete actions that are performed to the physical configuration of the production line.

2. *Operational knowledge:* This is knowledge about operators. Operators are actions that are executed under specific circumstances. Operators are invoked by strategies to achieve the specific goals specified in decision frames. Operators can be categorized concerning the context, that is the goal, in which they can be used:

   - Operators for reducing the blocking time of a machine. Such an operator is the following one:

     if the goal is 'reduce the blocking time' of a machine

then find a source buffer and transfer one buffer unit from the source buffer to the buffer next to the one with the maximum blocking time.

- Operators for allocating the total buffer size. An example of such an operator is:

  if the goal is 'allocation of the total buffer size' and the station with the maximum blocking time needs more buffer units,

  then allocate one buffer unit to the buffer next to the station with the maximum blocking time.

- Operators for increasing throughput:

  if the goal is to increase throughput and there is not a significant difference between the average delay times of two adjacent stations,

  then find a source buffer and transfer one unit to each buffer of the adjacent stations.

Operators, although specified as PROLOG rules in the prototype system developed, have the form: <*Preconditions, Effects*>.

Preconditions check for particular conditions in the physical configuration of the line.

Effects are procedures that affect the physical configuration. Such procedures allocate units to a buffer and transfer specific number of buffer units from one station to another.

3. *Control knowledge:* This kind of knowledge comprises the necessary mechanisms for exploitation of the knowledge categories described above. Particularly, the control mechanism

   - monitors the execution of strategies and operators,
   - searches for alternative ways of executing strategies and operators and
   - searches for alternative strategies and operators to satisfy the current goals.

The control task is a top-down one. Given a particular goal, the system determines what changes in the physical configuration should be done, in order the target goal to be achieved. The changes are made by particular operators invoked by specific strategies. The control task is implemented in PROLOG. The major steps performed by the control mechanism are the following:

(a) Checks the preconditions of a strategy.

(b) Instantiates the strategy by executing the strategic steps specified in the strategy body.

(c) During strategy instantiation, strategic decisions are recorded in the decision frames which drive and justify the execution of subsequent strategic steps.

(d) When the strategy has been executed, post conditions are checked. In case one of these fails, then alternative ways of executing strategic steps are tested.

(e) If all the alternatives have been checked, then the strategy instantiation fails and the strategic decisions recorded in the decision frame are withdrawn.

4. *Heuristic knowledge:* This kind of knowledge comprises mainly conditions that check for the occurrence of specific configurations in the production line. Such configurations justify deciding what changes should be made to a given buffer allocation plan.

   For example, heuristic rules are utilised in order ASBA2 to decide the source buffer (the one from which can get buffer units) and the target buffer (the one to which units shall be allocated). A major heuristic used in these cases is the following one:

   To increase throughput, reduce blocking times in stations.

4 Numerical results

In this section, we present numerical results that have been obtained by running ASBA2. Results are splitted into two classes: (a) for exponential lines and (b) for Erlangian lines.

Specifically, Tables 1 to 7, provide numerical results concerning the buffer allocation in production lines with exponentially distributed service and repairing rates, whereas, Tables 8 to 14, provide analogous numerical results for production lines with service and repairing rates following the Erlangian distribution.

Recall that the target goal of ASBA2 is to maximize the throughput of a production line. The achieved throughput is denoted by X_K in Tables 1-14 and is given by column 3. Respectively, column 5 gives the maximum throughput, i.e., the throughput achieved by the optimal buffer allocation. This is denoted by X_K^* and is given in column 4 . Columns 1 and 2 give, respectively, the total buffer size, denoted by N, and the near optimum buffer allocation obtained by the proposed generative method ASBA2, denoted by BA. Lastly, column 6 of Tables gives D, the absolute percentage difference between the throughput achieved from ASBA2 and the maximum throughput, i.e., $D = \frac{|X_K^* - X_K|}{X_K^*} \times 100$. This expresses the failure of our system to achieve the required throughput.

Although ASBA2 uses a simulation method to calculate the throughput of the production line, the throughput X_K of the near optimal buffer allocation (given in column 3) is calculated using the exact method developed by Heavey, Papadopoulos and Browne [13]. This allows having a standard method towards which to measure the failure of our system to achieve the maximum throughput.

ASBA2, as well as its predecessor ASBA, is able to solve the problem for large production lines, allowing large buffer capacities for each buffer. However, to have a measure of ASBA2 success, we have studied its behaviour for a number of cases whose optimal buffer allocation can be calculated using enumerative methods. Therefore, we have studied ASBA2's performance mainly for production lines (balanced and unbalanced) with a small number of stations and small number of total buffer units.

That is, for 3-station, 4-station and 5-station production lines. The first buffer in all cases is considered to be saturated (containing always more than 0 jobs).

Table 1: ASBA2's results in buffer allocation of 3-station *exponential* lines with $\mu_1 = 1, \mu_2 = 1.2, \mu_3 = 1.4, r_1 = r_2 = r_3 = 0.5, \beta_1 = 0.05, \beta_2 = 0.05, \beta_3 = 0.05$

N	BA	X_K	OBA	X_K^*	D
1	$(1-0)$	0.6341	$(1-0)$	0.6341	0.00
2	$(1-1)$	0.6744	$(1-1)$	0.6744	0.00
3	$(2-1)$	0.7113	$(2-1)$	0.7113	0.00
4	$(3-1)$	0.7361	$(3-1)$	0.7361	0.00
5	$(3-2)$	0.7587	$(3-2)$	0.7587	0.00
6	$(3-3)$	0.7716	$(4-2)$	0.7777	0.78
7	$(4-3)$	0.7911	$(5-2)$	0.7922	0.13
8	$(5-3)$	0.8060	$(5-3)$	0.8060	0.00
9	$(6-3)$	0.8178	$(6-3)$	0.8178	0.00
10	$(6-4)$	0.8266	$(7-3)$	0.8274	0.09

Table 2: ASBA2's results in buffer allocation of 3-station *exponential* lines with $\mu_1 = 1, \mu_2 = 1.2, \mu_3 = 1.4, r_1 = r_2 = r_3 = 0.5, \beta_1 = 0.05, \beta_2 = 0.01, \beta_3 = 0.25$

N	BA	X_K	OBA	X_K^*	D
1	$(1-0)$	0.5748	$(1-0)$	0.5748	0.00
2	$(1-1)$	0.6204	$(1-1)$	0.6204	0.00
3	$(1-2)$	0.6486	$(2-1)$	0.6493	1.10
4	$(2-2)$	0.6783	$(2-2)$	0.6783	0.00
5	$(3-2)$	0.6982	$(2-3)$	0.6987	0.07
6	$(2-4)$	0.7141	$(3-3)$	0.7188	0.65
7	$(4-3)$	0.7334	$(3-4)$	0.7343	0.12
8	$(3-5)$	0.7465	$(4-4)$	0.7489	0.32
9	$(4-5)$	0.7610	$(4-5)$	0.7610	0.00
10	$(5-5)$	0.7720	$(5-5)$	0.7720	0.00

Table 3: ASBA2's results in buffer allocation of 3-station *exponential* lines with $\mu_1 = \mu_2 = \mu_3 = 1, r_1 = r_2 = r_3 = 0.5, \beta_1 = 0.05, \beta_2 = 0.01, \beta_3 = 0.25$

N	BA	X_K	OBA	X_K^*	D
1	$(1-0)$	0.4700	$(0-1)$	0.4814	2.36
2	$(1-1)$	0.5137	$(1-1)$	0.5137	0.00
3	$(2-1)$	0.5306	$(1-2)$	0.5409	1.90
4	$(2-2)$	0.5583	$(1-3)$	0.5600	0.30
5	$(3-2)$	0.5687	$(2-3)$	0.5773	1.48
6	$(3-3)$	0.5878	$(2-4)$	0.5914	0.60
7	$(5-2)$	0.5801	$(2-5)$	0.6022	3.66
8	$(2-6)$	0.6109	$(3-5)$	0.6122	0.21
9	$(6-3)$	0.6028	$(3-6)$	0.6204	2.83
10	$(5-5)$	0.6232	$(3-7)$	0.6271	0.62

Table 4: ASBA2's results in buffer allocation of 4-station *exponential* lines with $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 1, r_1 = r_2 = r_3 = r_4 = 0.5, \beta_1 = 0.05, \beta_2 = 0.02, \beta_3 = 0.01, \beta_4 = 0.001$

N	BA	X_K	OBA	X_K^*	D
1	$(0-1-0)$	0.5250	$(0-1-0)$	0.5250	0.00
2	$(1-1-0)$	0.5577	$(1-1-0)$	0.5577	0.00
3	$(1-1-1)$	0.5916	$(1-1-1)$	0.5916	0.00
4	$(2-1-1)$	0.6128	$(1-2-1)$	0.6155	0.43
5	$(2-1-2)$	0.6307	$(2-2-1)$	0.6378	1.11
6	$(2-2-2)$	0.6563	$(2-2-2)$	0.6563	0.00
7	$(3-2-2)$	0.6721	$(2-3-2)$	0.6725	0.05
8	$(3-3-2)$	0.6890	$(3-3-2)$	0.6890	0.00
9	$(5-2-2)$	0.6899	$(3-4-2)$	0.7007	1.54
10	$(5-3-2)$	0.7076	$(4-3-3)$	0.7129	0.74

Table 5: ASBA2's results in buffer allocation of 4-station *exponential* lines with

$\mu_1 = 1.6, \mu_2 = 1.4, \mu_3 = 1.2, \mu_4 = 1, r_1 = r_2 = r_3 = r_4 = 0.5, \beta_1 = 0.05, \beta_2 = 0.02, \beta_3 = 0.01, \beta_4 = 0.005$

N	BA	X_K	OBA	X_K^*	D
1	$(0-1-0)$	0.6476	$(0-0-1)$	0.6572	1.46
2	$(1-1-0)$	0.6652	$(0-1-1)$	0.7064	5.83
3	$(1-1-1)$	0.7289	$(0-1-2)$	0.7361	0.97
4	$(2-1-1)$	0.7394	$(1-1-2)$	0.7673	3.63
5	$(1-3-1)$	0.7670	$(1-2-2)$	0.7938	3.37
6	$(1-1-4)$	0.8013	$(1-2-3)$	0.8173	1.95
7	$(2-3-2)$	0.8166	$(1-3-3)$	0.8346	2.15
8	$(3-2-3)$	0.8348	$(1-3-4)$	0.8514	1.95
9	$(4-2-3)$	0.8390	$(2-3-4)$	0.8660	3.11
10	$(3-2-5)$	0.8672	$(2-3-5)$	0.8796	1.41

Table 6: ASBA2's results in buffer allocation of 5-station *exponential* lines with

$\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = 1, r_1 = r_2 = r_3 = r_4 = r_5 = 0.5, \beta_1 = 0.1, \beta_2 = 0.2, \beta_3 = 0.25, \beta_4 = 0.3, \beta_5 = 0.35$

N	BA	X_K	OBA	X_K^*	D
1	$(1-0-0-0)$	0.2848	$(0-0-1-0)$	0.2997	4.97
2	$(2-0-0-0)$	0.2870	$(0-0-1-1)$	0.3158	9.12
3	$(2-1-0-0)$	0.3005	$(0-1-1-1)$	0.3340	10.00
4	$(1-1-1-1)$	0.3420	$(0-1-2-1)$	0.3471	1.47
5	$(1-2-1-1)$	0.3512	$(0-1-2-2)$	0.3579	1.87
6	$(2-2-1-1)$	0.3538	$(0-2-2-2)$	0.3699	4.35
7	$(3-1-1-2)$	0.3603	$(1-2-2-2)$	0.3794	5.03
8	$(2-2-2-2)$	0.3833	$(1-2-3-2)$	0.3894	1.56
9	$(3-2-2-2)$	0.3852	$(1-2-3-3)$	0.3990	3.45
10	$(3-1-3-3)$	0.3956	$(1-3-3-3)$	0.4066	2.70

Table 7: ASBA2's results in buffer allocation of 5-station *exponential* lines with

$\mu_1 = 1, \mu_2 = 1.1, \mu_3 = 1.2, \mu_4 = 1.3, \mu_5 = 1.4, r_1 = r_2 = r_3 = r_4 = r_5 = 0.5, \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0.05$

N	BA	X_K	OBA	X_K^*	D
1	$(0-1-0-0)$	0.5213	$(0-1-0-0)$	0.5213	0.00
2	$(1-1-0-0)$	0.5514	$(1-1-0-0)$	0.5514	0.00
3	$(1-1-1-0)$	0.5824	$(1-1-1-0)$	0.5824	0.00
4	$(1-1-1-1)$	0.5976	$(1-2-1-0)$	0.6027	0.84
5	$(2-1-1-1)$	0.6196	$(2-2-1-0)$	0.6213	0.27
6	$(2-2-1-1)$	0.6422	$(2-2-1-1)$	0.6422	0.00
7	$(1-2-2-2)$	0.6380	$(2-2-2-1)$	0.6585	3.11
8	$(1-3-2-2)$	0.6519	$(3-2-2-1)$	0.6744	3.33
9	$(3-2-2-2)$	0.6829	$(3-3-2-1)$	0.6894	0.94
10	$(2-4-2-2)$	0.6914	$(3-3-3-1)$	0.7005	1.30

Table 8: ASBA2's results in buffer allocation of 3-station *Erlangian* lines with $\mu_1 = 1, \mu_2 = 1.2, \mu_3 = 1.4, r_1 = r_2 = r_3 = 0.5, \beta_1 = 0.05, \beta_2 = 0.05, \beta_3 = 0.05$

N	BA	X_K	OBA	X_K^*	D
1	$(1-0)$	0.7083	$(1-0)$	0.7083	0.00
2	$(1-1)$	0.7489	$(1-1)$	0.7489	0.00
3	$(2-1)$	0.7857	$(2-1)$	0.7857	0.00
4	$(2-2)$	0.8026	$(3-1)$	0.8086	0.74
5	$(2-3)$	0.8109	$(3-2)$	0.8261	1.84
6	$(3-3)$	0.8347	$(4-2)$	0.8421	0.87
7	$(4-3)$	0.8507	$(5-2)$	0.8538	0.36
8	$(4-4)$	0.8555	$(6-2)$	0.8627	0.83
9	$(5-4)$	0.8669	$(6-3)$	0.8709	0.46
10	$(7-3)$	0.8776	$(7-3)$	0.8776	0.00

Table 9: ASBA2's results in buffer allocation of 3-station *Erlangian* lines with $\mu_1 = 1, \mu_2 = 1.2, \mu_3 = 1.4, r_1 = r_2 = r_3 = 0.5, \beta_1 = 0.05, \beta_2 = 0.01, \beta_3 = 0.25$

N	BA	X_K	OBA	X_K^*	D
1	$(1-0)$	0.6377	$(1-0)$	0.6377	0.00
2	$(1-1)$	0.6896	$(1-1)$	0.6896	0.00
3	$(1-2)$	0.7192	$(1-2)$	0.7192	0.00
4	$(3-1)$	0.7354	$(2-2)$	0.7473	1.60
5	$(3-2)$	0.7650	$(2-3)$	0.7670	0.26
6	$(3-3)$	0.7841	$(3-3)$	0.7841	0.00
7	$(4-3)$	0.7961	$(3-4)$	0.7978	0.21
8	$(4-4)$	0.8092	$(4-4)$	0.8092	0.00
9	$(6-3)$	0.8124	$(4-5)$	0.8191	0.81
10	$(5-5)$	0.8273	$(5-5)$	0.8273	0.00

Table 10: ASBA2's results in buffer allocation of 3-station *Erlangian* lines with $\mu_1 = \mu_2 = \mu_3 = 1, r_1 = r_2 = r_3 = 0.5, \beta_1 = 0.05, \beta_2 = 0.01, \beta_3 = 0.25$

N	BA	X_K	OBA	X_K^*	D
1	$(1-0)$	0.5177	$(0-1)$	0.5349	3.22
2	$(1-1)$	0.5678	$(1-1)$	0.5678	0.00
3	$(1-2)$	0.5947	$(1-2)$	0.5947	0.00
4	$(3-1)$	0.5892	$(1-3)$	0.6119	3.71
5	$(3-2)$	0.6159	$(2-3)$	0.6247	1.41
6	$(4-2)$	0.6203	$(2-4)$	0.6354	2.37
7	$(4-3)$	0.6356	$(2-5)$	0.6429	1.13
8	$(4-4)$	0.6452	$(2-6)$	0.6483	0.47
9	$(5-4)$	0.6475	$(3-6)$	0.6529	0.83
10	$(5-5)$	0.6534	$(3-7)$	0.6563	0.44

Table 11: ASBA2's results in buffer allocation of 4-station *Erlangian* lines with $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 1, r_1 = r_2 = r_3 = r_4 = 0.5, \beta_1 = 0.05, \beta_2 = 0.02, \beta_3 = 0.01, \beta_4 = 0.001$

N	BA	X_K	OBA	X_K^*	D
1	$(1-0-0)$	0.5936	$(0-1-0)$	0.6024	1.46
2	$(2-0-0)$	0.6074	$(1-1-0)$	0.6407	5.20
3	$(1-1-1)$	0.6792	$(1-1-1)$	0.6792	0.00
4	$(2-0-2)$	0.6608	$(1-2-1)$	0.7019	5.86
5	$(3-0-2)$	0.6720	$(2-2-1)$	0.7258	7.41
6	$(2-2-2)$	0.7422	$(2-2-2)$	0.7422	0.00
7	$(3-2-2)$	0.7580	$(3-2-2)$	0.7580	0.00
8	$(4-2-2)$	0.7680	$(3-3-2)$	0.7723	0.56
9	$(4-2-3)$	0.7776	$(4-3-2)$	0.7828	0.66
10	$(4-3-3)$	0.7926	$(4-3-3)$	0.7926	0.00

Table 12: ASBA2's results in buffer allocation of 4-station *Erlangian* lines with

$\mu_1 = 1.6, \mu_2 = 1.4, \mu_3 = 1.2, \mu_4 = 1, r_1 = r_2 = r_3 = r_4 = 0.5, \beta_1 = 0.05, \beta_2 = 0.02, \beta_3 = 0.01, \beta_4 = 0.005$

N	BA	X_K	OBA	X_K^*	D
1	$(0-0-1)$	0.7512	$(0-0-1)$	0.7512	0.00
2	$(0-0-2)$	0.7780	$(0-1-1)$	0.8029	3.10
3	$(1-1-1)$	0.8263	$(0-1-2)$	0.8339	0.91
4	$(3-0-1)$	0.7914	$(1-1-2)$	0.8627	8.26
5	$(2-2-1)$	0.8512	$(1-1-3)$	0.8838	3.69
6	$(2-2-2)$	0.8919	$(1-2-3)$	0.9056	1.51
7	$(3-1-3)$	0.9002	$(1-2-4)$	0.9199	2.14
8	$(2-4-2)$	0.9068	$(1-3-4)$	0.9315	2.65
9	$(3-2-4)$	0.9342	$(1-3-5)$	0.9415	0.78
10	$(4-2-4)$	0.9367	$(2-3-5)$	0.9497	1.37

Table 13: ASBA2's results in buffer allocation of 5-station *Erlangian* lines with

$\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = 1, r_1 = r_2 = r_3 = r_4 = r_5 = 0.5, \beta_1 = 0.1, \beta_2 = 0.2, \beta_3 = 0.25, \beta_4 = 0.3, \beta_5 = 0.35$

N	BA	X_K	OBA	X_K^*	D
1	$(1-0-0-0)$	0.3192	$(0-0-1-0)$	0.3378	5.51
2	$(1-1-0-0)$	0.3337	$(0-0-1-1)$	0.3578	6.74
3	$(2-1-0-0)$	0.3348	$(0-1-1-1)$	0.3782	11.50
4	$(1-1-1-1)$				
5	$(2-1-1-1)$				
6	$(3-1-1-1)$				
7	$(3-2-1-1)$				
8	$(3-2-1-2)$				
9	$(3-2-2-2)$				
10	$(4-2-3-3)$				

Table 14: ASBA2's results in buffer allocation of 5-station *Erlangian* lines with

$\mu_1 = 1, \mu_2 = 1.1, \mu_3 = 1.2, \mu_4 = 1.3, \mu_5 = 1.4, r_1 = r_2 = r_3 = r_4 = r_5 = 0.5, \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0.05$

N	BA	X_K	OBA	X_K^*	D
1	$(0-0-1-0)$	0.5825	$(0-1-0-0)$	0.5968	2.40
2	$(1-0-1-0)$	0.6265	$(1-1-0-0)$	0.6338	1.15
3	$(1-1-1-0)$	0.6673	$(1-1-1-0)$	0.6673	0.00
4	$(1-1-1-1)$				
5	$(1-1-2-1)$				
6	$(2-1-1-2)$				
7	$(3-1-2-1)$				
8	$(1-3-2-2)$				
9	$(2-3-2-2)$				
10	$(3-2-3-2)$				

Remarks: From the results given in Tables 1-14, we can make the following observations:

1. Results produced by ASBA2 are very close to the optimal solutions. However, we sacrifice the optimal solution for computational efficiency: ASBA systems do not need to enumerate all the possible solutions for producing the near-optimum solution. They rather exploit specific knowledge about allocating buffer storage. Particularly, as it can be noticed, ASBA2's failure to achieve the maximum throughput in the cases where service and repairing rates are considered to be exponential is in average close to 1.43% and only in 4 cases out of 70, in Tables 5 and 6, its measure of failure exceeds 5%.

   Moreover, ASBA2's failure to achieve the maximum throughput in cases where service and breakdown rates are considered to be Erlang is in average close to 1.73% and in 7 cases out of 56, in Tables 11-13, its measure of failure exceeds 5%. We must notice that in Tables 13 and 14, ASBA2 has succeeded in calculating a near optimal buffer allocation, in cases where exact techniques have failed to provide the optimal buffer allocation. We should also notice that for these cases, the exact throughput of the production line for the near opimal buffer allocation can not be calculated by the exact methods utilised.

2. Calculations provided by ASBA2 can be explained in terms of the strategies, operational and heuristic knowledge used. This enables us to investigate the utilisation of various heuristic rules which intuitively seem promising in calculating optimal buffer allocations. However, since ASBA2 is combined with a simulation method, it must be stated that in specific cases, promising rules fail, due to simulation weakness to provide results that justify making progress towards maximizing throughput. This is one of the major drawbacks of ASBA2.

5 Concluding remarks

The paper describes ASBA2, a knowledge based system that solves the well-known problem for buffer allocation in production lines. The system is an extension of ASBA, which authors have described in an earliest paper. ASBA allocates buffer space in reliable production lines, aiming at reducing WIP, subject to a given total buffer space and a required throughput. Although ASBA computes near optimal buffer allocations for reliable, balanced and unbalanced, production lines, ASBA2 aims to extend the functionality of ASBA to unreliable, balanced and unbalanced, production lines. The paper describes how ASBA2 computes a near optimal buffer allocation for unreliable lines, aiming at increasing throughput subject to given total buffer space. The paper gives results for specific cases where the optimal buffer allocation is known and draws conclusions for ASBA2's behaviour. Results, compared with exact methods, are encouraging. The system, in a computationally efficient way, using specific types of knowledge, computes results very close to the optimal ones.

We plan to extend ASBA2 to deal with WIP as well, i.e., with the problem that ASBA dealt, but for unreliable production lines. However, our first step in this direction will be to design a simulation algorithm that computes performance measures concerning the behaviour of a production line that do not deviate much from computations that would be given from exact methods.

References

[1] R. Suri, An overview of evaluative models for flexible manufacturing systems. *Annals of Operations Research*, **3**, 115– (1985).

[2] H. T. Papadopoulos, C. Heavey and J. Browne, *Queueing Theory in Manufacturing Systems Analysis and Design*. Chapman and Hall, London (1993).

[3] H. T. Papadopoulos and G. A. Vouros, A model management system (MM-S) for the design and operation of production lines. *International Journal of Production Research*, In press (1997).

[4] H. T. Papadopoulos and C. Heavey, Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines. *European Journal of Operational Research*, **92**, 1–27 (1996).

[5] R. W. Conway, W. L. Maxwell, J. O. McClain and L. J. Thomas, The role of work-in-process inventories in series production lines. *Operations Research*, **36**, 229–241 (1988).

[6] F. S. Hillier and K. C. So, The effect of the coefficient of variation of operation times on the allocation of storage space in production line system. *IIE Transactions*, **23**, 198–206 (1991).

[7] F. S. Hillier, K. C. So and R. W. Boling, Notes: Toward characterizing the optimal allocation of storage space in production line systems with variable processing times. *Management Science*, **39**(1), 126–133 (1993).

[8] S. G. Powell, Buffer allocation in unbalanced serial lines. Working Paper No. 289, The Amos Tuck School of Business Administration, Dartmouth College (1992).

[9] K. C. So, Optimal buffer allocation strategy for minimizing work-in-process inventory in unpaced production lines. Working Paper, Presented at the ORSA/TIMS meeting in Detroit (USA), October (1994).

[10] H. T. Papadopoulos and M. I. Vidalis, Optimal buffer allocation in reliable production lines: (I) The balanced case. Submitted for publication (1996a).

[11] H. T. Papadopoulos and M. I. Vidalis, Optimal buffer allocation in 'semi-balanced' unreliable production lines. Submitted for publication (1996b).

[12] H. T. Papadopoulos and M. I. Vidalis, An optimization algorithm for the buffer allocation in unreliable unbalanced production lines. Submitted for publication (1996c).

[13] C. Heavey, H. T. Papadopoulos and J. Browne, The throughput rate of multi-station unreliable production lines. *European Journal of Operational Research*, **68**, 69–89 (1993).

[14] H. T. Papadopoulos and M. I. Vidalis, Some results on the effective buffer management for minimizing WIP inventory in production lines: (I) The reliable case. Submitted for publication (1996d).

[15] G. A. Vouros, T. Panayiotopoulos and C. D. Spyropoulos, A framework for developing expert loading systems for product carriers. *Expert Systems with Applications, an International Journal*, **10**(1) (1996).