

**CONTROL OF AN ASSEMBLY SYSTEM
WITH PROCESSING TIME AND
SUBASSEMBLY TYPE UNCERTAINTY**

Matthew F. Kebulis
and
Izak Duenyas
Department of Industrial & Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109-2117

Technical Report 95-9

August 1995

CONTROL OF AN ASSEMBLY SYSTEM WITH PROCESSING TIME AND SUBASSEMBLY TYPE UNCERTAINTY

MATTHEW F. KEBLIS and IZAK DUENYAS

Department of Industrial and Operations Engineering

University of Michigan, Ann Arbor, MI, 48109

Abstract

We address the problem of controlling an assembly system in which the processing times as well as the “types” of subassemblies are stochastic. The quality (or performance) of the final product depends upon the characteristics of the subassemblies to be assembled, which are not constant. Furthermore, the processing time of a subassembly is random. We analyze the tradeoff between the increase in the potential value of products gained by delaying the assembly operation, and the inventory costs caused by this delay. We also consider the effects of processing time uncertainty. Our problem is motivated by the assembly of passive and active plates in flat panel display manufacturing.

We formulate the optimal control problem as a Markov Decision Process. However, the optimal policy is very complex, and we therefore develop simple heuristic policies. We report the results of a simulation study which tests the performance of our heuristics. The computational results indicate that the heuristics are effective for a wide variety of cases.

1 Introduction

Assembly systems, consisting of several sub-assembly lines feeding an assembly station, are prevalent in many manufacturing environments. Two typical examples in electronics manufacturing include multi-plane circuit boards (PCB's), which are manufactured by fabricating the layers separately and then laminating them together, and flat panel displays, where “active” and “passive” layers of an electronic display are produced separately and then mated.

Previous work on assembly systems has focused on the effects of the uncertainty of the processing times at the subassembly and assembly stages (e.g., Ammar (1980), Bhat (1986), Bonomi (1987), Lee and Pollock (1989), Hopp and Simon (1989), Duenyas and Hopp (1992, 1993), Duenyas (1994), Duenyas

and Kebulis (1995), Rao and Suri (1994), and Hazra and Seidmann(1994). The focus in this body of work has been the development of approximations for the performance of different release control mechanisms for assembly systems.

A significant other source of uncertainty in some assembly systems is “type uncertainty” where the “type” of subassemblies produced by one or more of the subassembly lines or machines is uncertain. In all of the above papers, it is assumed that only a single “type” of product is produced by each subassembly line. In contrast, in this paper, we focus on environments where there is uncertainty with respect to the “type” of subassembly to be produced. Furthermore, the performance (or quality) of the final product depends upon the characteristics or “types” of the subassemblies mated. In such an environment, the major tradeoff is between the better performance obtained by putting off the mating operation until a “match” of components with the same “types” is obtained, and the larger inventories associated with this delay in the mating operation.

A typical example of assembly systems with processing time as well as “type” uncertainty is in flat panel display manufacturing where “active” and “passive” layers of an electronic display are produced by separate processes. Due to yield losses, machine failures etc., the time between the production of two consecutive “active” or “passive” layers is random so that at any point in time there may be more “active” than “passive” layers or vice versa. After a passive and active layer are mated, the resulting electronic sandwich is cut into smaller pieces to produce final products (i.e., displays). The final product is defective and will have to be discarded if either the active or passive layer has a fatal defect in the location corresponding to that particular display. For this reason, each “passive” and “active” layer is inspected for fatal defects before the mating of layers and identified as being one of a set of possible “types”, each containing defects in specific locations. For example, when the layers will eventually be cut into 8 pieces (as is common in industrial applications), there are 2^8 types, since each of the 8 possible pieces may have zero or more fatal defects. (We note that the problem would be much simpler if the active and passive layers were first cut into smaller pieces and then only the non-defective pieces mated. However, under the currently available technology for flat panel display manufacturing, cutting the layers into smaller pieces before mating them significantly increases the defect rates. Therefore, the layers are first inspected, then mated, and then cut into smaller pieces).

In the flat panel display environment described above, the control issues to be addressed are as follows: Given the number and “types” of passive and active layers already produced and not yet mated 1) how does one decide when to release new passive or active layers into the system? 2) how does one decide

which active or passive layers (if any) to mate?

A similar problem arises in the manufacturing of ball bearings where an inner and outer race are assembled (along with a set of balls) to produce a ball bearing (Iyama et al. 1992). Each race has a critical dimension that is a random variable, taken from a known distribution. After the races are produced, they are accurately measured at an inspection machine and classified into one of three size ranges (e.g., types). High quality bearings are produced by mating (i.e., assembling), when possible, inner and outer races having the same size range. The “type” uncertainty is due to the variability in the dimensions of the races, and the processing time uncertainty is due to the possibility that the machines producing the inner and outer races may fail. For this problem, Iyama et al. investigate the effects of an ad-hoc mating strategy on buffers and machine blocking. They do not focus on the derivation of “optimal” mating strategies.

To our knowledge, the only paper that addresses the issue of control of assembly systems under type uncertainty is by Duenyas et al. (1994). However, Duenyas et al. assume that there is no processing time uncertainty and that the subassembly lines produce at the same deterministic rate. Therefore, the only decision in their paper is which (if any) subassemblies to mate. Clearly, in most realistic systems, processing times are not deterministic. Furthermore, even in the rare situation where they are deterministic, it is highly unlikely that all subassembly machines (or lines) produce at the exact same rate. For this reason, in this paper, we address the control problem when there is uncertainty in both the subassembly processing times and types.

The rest of this paper is organized as follows. Section 2 presents the problem formulation and introduces notation. In Section 3, we formulate the optimal control problem as a Markov Decision Process (MDP). Since the dynamic programming approach used for computing the optimal policy suffers from the “curse of dimensionality” when the number of types is greater than 3, we present simple heuristics in Section 4. Section 5 presents a comparison of our heuristics against simulation for sample problems with three and 16 types. Section 6 concludes the paper.

2 Problem Formulation and Notation

We restrict ourselves to the case where two components are produced, and combined to form a final product. For convenience, we refer to these as “left” and “right” halves, respectively. The act of combining halves is called “mating”. Each half is produced on a separate machine. (In this paper, we assume that each subassembly is produced on a single machine. Clearly, in many cases, subassemblies may be processed

on a tandem line. We intend to analyze more complicated network structures in the future.) We assume that the processing time distribution of the machine that produces the left (right) half has mean $\frac{1}{\mu_1}$ ($\frac{1}{\mu_2}$) and variance σ_1^2 (σ_2^2). At any point in time, a machine is either on (running) or off (stopped).

When a left half is produced, it exhibits a “type” $t \in \{1, 2, \dots, T\}$, with probability l_t ; type u right halves are produced with probability r_u , where $\sum_{t=1}^T l_t = 1$, and $\sum_{u=1}^T r_u = 1$. The probability that a right (left) half is of a certain type is independent of the type of previously produced left halves and right halves. In order to measure the negative effect of holding inventory and long cycle times, we assume that each half kept in inventory incurs a cost at rate h per unit of time.

When a right half of type u is mated with a left half of type t , the resulting product has a value V_{tu} , where

$$V_{tu} > 0 \tag{2.1}$$

$$V_{tt} > V_{tu} \text{ for } t \neq u \text{ and } V_{uu} > V_{tu} \text{ for } t \neq u \tag{2.2}$$

Inequality (2.1) implies that all matings have some value; (2.2) shows that mating left and right halves of the same type produces a maximum value. (For convenience, we define a “match” to be the mating of two halves of the same type.)

When there are three or more types, we will also assume the V_{tu} satisfy the additional inequality

$$V_{tt} + V_{uz} \geq V_{ut} + V_{tz} \quad \text{for all } z \neq u, u \neq t, z \neq t \tag{2.3}$$

This inequality ensures that immediately matching two halves of the same “type” is optimal. To see this, suppose that two halves of the same type t were held in inventory without being matched. This would imply that they would eventually be mated with some other halves (e.g., with a left half of type u and a right half of type z). However, by (2.3) this mating would result in no greater value than matching the two halves of type t and mating the two other halves of types u and z . Since there is a cost associated with holding inventory, it would be better to match the two halves of type t as soon as possible rather than hold them.

Inequalities (2.1) through (2.3) rule out some possible value matrices, however, they are reasonable for a wide variety of situations including the problem that motivated this study. In flat panel display manufacturing, the highest value is obtained when layers that have defects at the same locations are mated. This is because once mated, a location is defective if it has defects on *either* layer and the display corresponding to that location will have to be discarded. Each layer can be represented by a vector of zeros

and ones that denote whether a location is, respectively, defective or nondefective (e.g., $(1, 0)$ represents a layer that will be divided into two pieces with the second piece defective and the first piece nondefective). Let \underline{x} denote the vector associated with a passive or active layer. The unique type number $t(\underline{x})$ can then be represented as $t(\underline{x}) = \sum_{i=1}^N 2^i x_i$, with N denoting the number of pieces that the layer will be cut into. Conversely, given type number t , the vector $\underline{x}(t)$ is uniquely defined to be binary representation of t . We can similarly represent the right layer by a vector \underline{y} , with type $u(\underline{y})$. The value of mating/matching a left half of type t with a right half of type u is then given by $V_{tu} = r_1(\underline{x}(t) \cdot \underline{y}(u)) + r_2(N - (\underline{x}(t) \cdot \underline{y}(u)))$, where r_1 is the revenue associated with a good display, r_2 is the salvage value of a defective display and $(\underline{x}(t) \cdot \underline{y}(u))$ is the inner-product of vector $\underline{x}(t)$ and $\underline{y}(u)$. It is straightforward to check that when $r_1 \geq r_2$ this value function satisfies all three inequalities. (Duenyas et al., 1994).

Since an optimal policy immediately matches left halves and the right halves of the same type, if the number of left halves of a certain type is non-zero, then the number of right halves must be zero. Therefore, for each type, we only need to keep track of the difference between the number of left halves and the number of right halves. The state of the system can be represented as the T -vector, $\{n_1, \dots, n_T\}$, where n_t is the difference between the number of left halves of type t and right halves of type t . The fundamental problem is to determine for any given state vector (n_1, n_2, \dots, n_T) , which pair (if any) of types should be mated and which machine(s) to run (left, right or both) in order to maximize net value per unit time. (We note that in the particular application that motivated this problem, the actual assembly process is much faster than the production of the subassemblies. We therefore ignore any queues at the assembly machine and focus on the inventory of parts waiting to be mated or matched).

3 Optimal Policy

In this section we give a Markov Decision Process (MDP) formulation for the problem described in Section 2 when processing times at both subassembly machines are assumed to be exponential. When processing times are non-exponential, one needs to keep track of the time since the last job at each machine was begun, and therefore, the formulation becomes even more complicated. We therefore focus on the case where processing times are exponential. As we will show, the formulation becomes quickly intractable even with exponential processing times. We will therefore develop a heuristic in the next section which does not assume exponential processing times.

We let

g = the minimum long-run cost per period

n = a vector which depicts the state of the system (n_t is the difference between the number of left halves of type t and the number of right halves of type t).

f_n = the relative value of being in state n

e_t = the unit vector whose t^{th} component is one.

We use uniformization as in Lippman (1975) and without loss of generality, we also assume that $\mu_1 + \mu_2 = 1$. When this does not hold true, appropriate rescaling of the problem can result in an equivalent problem with $\mu_1 + \mu_2 = 1$. The formulation consists of two cases: when all $n_t \geq 0$ for all $t = 1, \dots, T$ or $n_t \leq 0$ for all $t = 1, \dots, T$, the only option is to stop one of the machines (since in this case there are no parts to mate from one of the machines) and the underlying recursive equation is

$$g + f_n = (h\|n\| + \mu_1 \min\{\sum_{t=1}^T l_t(f_{n+e_t} - V_{tt} \cdot 1_{\{n_t < 0\}}); f_n\} + \mu_2 \min\{\sum_{t=1}^T r_t(f_{n-e_t} - V_{tt} \cdot 1_{\{n_t > 0\}}); f_n\}) \quad (3.4)$$

otherwise, we have

$$g + f_n = \min \left\{ \begin{array}{l} h\|n\| + \mu_1 \min\{\sum_{t=1}^T l_t(f_{n+e_t} - V_{tt} \cdot 1_{\{n_t < 0\}}), f_n\} + \mu_2 \min\{\sum_{t=1}^T r_t(f_{n-e_t} - V_{tt} \cdot 1_{\{n_t > 0\}}), f_n\} \\ \min_{(u,z) \in X} \left\{ \begin{array}{l} h(\|n\| - 2) - V_{uz} + \mu_1 \min\{\sum_{t=1}^T l_t(f_{n-e_u+e_z+e_t} - V_{tt} \cdot 1_{\{n_t+1_{\{t=z\}} < 0\}}), f_{n-e_u+e_z}\} \\ + \mu_2 \min\{\sum_{t=1}^T r_t(f_{n-e_u+e_z+e_t} - V_{tt} \cdot 1_{\{n_t-1_{\{t=u\}} > 0\}}), f_{n-e_u+e_z}\} \end{array} \right. \end{array} \right. \quad (3.5)$$

where $X = \{(u, z) : n_u > 0, n_u \cdot n_z < 0 \text{ and } u, z = 1, \dots, T\}$, 1_y the indicator function that equals 1 when y is true and 0 otherwise, and $\|n\| = |n_1| + |n_2| + \dots + |n_T|$. In equation (3.4), the term involving μ_1 pertains to the decision of either running or stopping the left machine. The term involving μ_2 pertains to the decision of either running or stopping the right machine. When a new component is produced, if it is a left half of type t , then the state becomes $n + e_t$ and a value V_{tt} is earned if there is a right half of type t in inventory. If the component produced is a right half of type t , then the state becomes $n - e_t$ and a value V_{tt} is earned if there is a left half of type t in inventory. In equation (3.5), the outside minimum pertains to the decision of either not mating any halves, or mating a left half of type u with a right half of type z . Within the outside minimum, the top line is the same as equation (3.4) and therefore has the same explanation. The bottom line is the minimum of all possible matings of left halves of type u with right halves of type z .

Equations (3.4) and (3.5) completely define the MDP formulation for the T-type problem. Unfortunately, the optimal solution to the MDP in (3.4) and (3.5) has an extremely complicated structure. The decision to shut off one of the machines, for example, is dependent on the number and types of left and right halves, and not only on the difference between the number of left and right halves. Moreover, as Duenyas et al. show, even for the case where there is no processing time uncertainty, the optimal policy becomes *very* complex. In fact, even if the optimal solution to equations (3.4) and (3.5) were available,

it would be questionable whether this solution would be implemented, as a huge database keeping the optimal decisions for each possible state would need to be stored and consulted after the production of each part. These considerations lead us to the development of two simple but effective heuristics which we describe in the next section.

4 Heuristic Solutions

The MDP formulation of the type matching problem quickly suffers from the “curse-of-dimensionality” as the number of types increases. For example, when there are 4 types, formulating an MDP where the number of left or right sides of a certain type is at most 40 requires over 40,000,000 states. Therefore, for realistically sized problems (for example, with 16 types) solving the MDP is not a practical solution. Furthermore, even if the solution were somehow available, a database that would store the optimal decision in each possible state would be infeasible to construct. This leads us to consider two heuristic solutions. The first heuristic (H1) is similar to one that we observed in use at a flat panel display manufacturing plant when we became aware of this problem. This heuristic decomposes the problem into two separate problems. The first problem is when to stop the left (or right) machine, and the second problem is how to decide whether or not to mate the existing parts and which ones to mate. Heuristic 1 applies two simple thresholds to address these problems. The first threshold, a_s , is the maximum number of left (or right) halves that are allowed to accumulate in inventory. If the number of left (right) halves in inventory reaches a_s , then the left (right) machine is stopped until the number of left (right) halves in inventory drops to $a_s - 1$. The second threshold, a_i , is the maximum number of total left and right halves that are allowed to accumulate in inventory. If the sum of left and right halves reaches a_i , then a transportation problem is solved to determine how to mate all the available halves optimally. The best values of a_s and a_i are determined through simulation for a given problem.

H1 is implemented as follows. Every time a left (right) half is produced, it is matched with a right (left) half of the same type if there are any in inventory. If a half of the same type is not available, then the newly produced half is added to inventory. If the sum of the left and right halves has reached a_i , then a transportation problem is solved to decide how to mate the available halves. In the case where the number of left and right halves available is not equal, the solution to the transportation problem will prescribe that some of the halves go unmated and remain in inventory. The levels of left and right halves in inventory are then compared to a_s , (regardless of whether a transportation problem has been solved to mate parts) in order to determine whether either machine should be stopped.

Despite its simplicity and ease of implementation, H1 has significant weaknesses. First of all, both thresholds have to be computed by simulation, which is time-consuming. Furthermore, a new simulation study is required if any of the system parameters is changed. Also, software for solving the transportation problem dynamically is required to decide which parts to mate (although in the computerized environment of flat panel display manufacturing, this is not as significant a problem). These problems with H1 lead us to develop a second heuristic, H2.

Our second heuristic (H2) also decomposes the combined problem of deciding when to shut off machines and mate parts to separate problems. However, no simulations are required to implement H2. To decide when to shut off the machines, we replace the original problem with T types by one with a single “average” type. In a problem with only one type of product, as soon as there is a single left and right half, these can immediately be assembled. Therefore, the only decision here is on when to shut off the left or right machine. The following theorem states the structure of the optimal policy for deciding when to shut off the machines for a 1-type problem.

Theorem 1 *The optimal policy for the 1-type assembly control problem where the machine processing times have exponential distributions is a control-limit policy requiring only the state n (a scalar, the difference between the number of left and right halves) and two numbers $a_l \geq 0$ and $a_r \leq 0$. The optimal policy is to:*

- a) stop the left machine if $n > a_l$;
- b) stop the right machine if $n < a_r$;
- c) wait for the next component produced if $a_r \leq n \leq a_l$.

Proof: The proof is straightforward and therefore omitted.

We replace the original T-type problem by a 1-type problem with the same holding cost, and machine processing times as the original problem. We assume that every time two parts are mated, a revenue of $R = \frac{\sum_{i=1}^T l_i r_i V_{ii}}{\sum_{i=1}^T l_i r_i}$ is earned. Note that R is an average revenue computed over the maximum fraction of parts that can be matched. We therefore replace the original T-type problem by a single type problem with the average revenue R per part. Given R , h and the machine processing time distributions, we can compute the two thresholds a_l , and a_r by solving the following simple dynamic program:

$$g + v_i = \frac{1}{\tau}(2h|i| + \mu_1 \min\{v_{i+1} - R1_{\{i < 0\}}; v_i\} + \mu_2 \min\{v_{i-1} - R1_{\{i > 0\}}; v_i\}). \quad (4.6)$$

In (4.6), g denotes the optimal cost per unit time, v_i denotes the relative cost of state i , and $\tau = \mu_1 + \mu_2$. With probability μ_1/τ , the next event is a completion of a left half, and the first minimization represents the choice between producing another left half (and getting a revenue R if there are only right halves in inventory) or not producing another left half by shutting off the machine. The explanation for the second minimum is similar.

Once the threshold values for shutting off either the left or the right machine are obtained, our decomposition heuristic also computes threshold values a_{tu} for mating a left half of type t with a right half of type u , for all $t = 1, \dots, T$ and $u = 1, \dots, T, u \neq t$. To do this, we decompose the original T-type problem to simpler 2-type problems. We solve 2-type problems (by ignoring the existence of the other types) to obtain the thresholds for mating those two types. For this, the production probabilities first have to be conditioned on producing only those particular types t and u ,

$$l'_t \equiv \Pr\{\text{left type } t \text{ produced} \mid \text{only left } t \text{ or } u \text{ produced}\} = \frac{l_t}{l_t + l_u} \equiv 1 - l'_u \quad (4.7)$$

$$r'_t \equiv \Pr\{\text{right type } t \text{ produced} \mid \text{only right } t \text{ or } u \text{ produced}\} = \frac{r_t}{r_t + r_u} \equiv 1 - r'_u \quad (4.8)$$

We also need to re-scale the holding costs. In an actual 2-type mating problem, the expected time until a new left (right) half arrives is $\frac{1}{\mu_1}(\frac{1}{\mu_2})$, when the left (right) machine is not stopped. With more than 2 types, the expected time until the arrival of another right half of type t or u equals $\frac{1}{\mu_2(r_t+r_u)}$. Thus, each left half of type t or u in inventory incurs an expected holding cost of $\frac{h}{\mu_2(r_t+r_u)}$ until the arrival of the next right half of either type. Similarly, each right half of type t or u incurs an expected cost $\frac{h}{\mu_1(l_t+l_u)}$ until the arrival of the next left half of either type. Thus, when we rescale the probabilities for types t and u , we also need to rescale the holding cost values. We assign the rescaled holding cost value $h' \equiv \frac{h}{2(l_t+l_u)} + \frac{h}{2(r_t+r_u)}$ to be the average of these two values.

Given the conditional probabilities, l'_t, r'_t, l'_u, r'_u , the rescaled holding cost h' , the values $\mu_1, \mu_2, V_{tu}, V_{ut}, V_{tt}$ and V_{uu} from the original problem and the thresholds for shutting off the left and right machines a_l and a_r , we can compute the optimal thresholds a_{tu} and a_{ut} for mating a half of t and a half of u . To do this, note that given the values of a_{tu} and a_{ut} , the system can be modelled as a simple Markov-Process for which we can compute the average cost per unit time. In this Markov Process, the state of the system is (i, j) where i (j) denotes the difference between the number of left and the number of right halves of type t (u). Except at the boundaries, (defined by a_l, a_r, a_{tu} and a_{ut}), the possible transitions from (i, j) are to states 1) $(i + 1, j)$ with rate $\mu_1 l'_t$ corresponding to the arrival of a left-half of type t , 2) $(i, j + 1)$ with rate $\mu_1 l'_u$ corresponding to the arrival of a left-half of type u , 3) $(i - 1, j)$ with rate $\mu_2 r'_t$ corresponding

to the arrival of a right-half of type t and 4) $(i, j - 1)$ with rate $\mu_2 r'_u$ corresponding to the arrival of a right-half of type u . Similarly, except at the boundaries, in state (i, j) costs are incurred with rate

$$h(|i| + |j|) - \mu_2(r'_t V_{tt} 1_{\{i>0\}} + r'_u V_{uu} 1_{\{j>0\}}) - \mu_1(l'_t V_{tt} 1_{\{i<0\}} + l'_u V_{uu} 1_{\{j<0\}}).$$

The steady state probabilities as well as the average cost per unit time for this Markov Process can be easily computed and the optimal values of a_{tu} and a_{ut} can be found by pure enumeration.

Once the thresholds a_l and a_r for shutting off the left and right machines and the thresholds for mating, a_{tu} and a_{ut} for all $t = 1, \dots, T$ and $u = 1, \dots, T$, have been computed, the implementation of H2 is straightforward. Every time a new left (right) half is produced, it is mated with a right (left) half of the same type if there are any in inventory. If a half is not available, then the newly produced half is added to inventory. For any $t = 1, \dots, T$ and $u = 1, \dots, T$, if there are at least a_{tu} left halves of type t and right halves of type u , then a left side of type t is mated with a right side of type u . After carrying out any matching or mating, the number of left and right halves in inventory are then compared to a_l and a_r respectively in order to determine whether either machine should be stopped.

H2 is at least as simple to implement in practice as H1. Furthermore the thresholds can be computed very quickly. For problems with as many as 16 types, (typical in the flat panel display environment) the computation of *all* of the thresholds (which have to be computed only once) takes less than an hour of CPU time on a Sun Sparcstation 20. Checking to see if any threshold has been exceeded is also very simple, especially in computerized electronics manufacturing environments.

Finally, we note that although we described both heuristics for the case where the processing times are assumed to be exponential, they can easily be adapted to the case where processing times are non-exponential. In particular, as the optimal thresholds for H1 are computed by simulation, whether the processing times are exponential or not makes no difference. In the case of H2, approximating the processing times by appropriate phase-type distributions results in Markov-chains for which the optimal thresholds can once again be computed, albeit at a cost of larger state spaces.

5 Computational Results

We conducted a simulation study to test the performance of the two heuristics. Since problems with more than 3 types become extremely difficult to solve optimally (as we noted above, an MDP formulation of a problem with 4 types can easily require over 40 million states) the two heuristics were tested against the

optimal policy only for problems with 3 types. We also compared the two heuristics for typical 16-type problems arising in flat panel display manufacturing.

Tables 1 through 3 show how the two heuristics performed on 36 test problems with 3 types. The results in Table 1 are for systems where both machines have exponential processing time distributions with rate 0.50. In Table 2, the results are for systems where each left machine has an exponential processing time distribution with rate 0.60 and each right machine has an exponential processing time distribution with rate 0.40. In Table 3, the results are for systems where each left machine has an exponential processing time distribution with rate 0.66 and each right machine has an exponential processing time distribution with rate 0.33. For each use of H1, simulation was used to compute the profit maximizing threshold values. Tables 1 through 3 show the average profit per unit time obtained by H1, H2 and the optimal policy (obtained by solving the complete MDP formulation), as well as the percentage sub-optimality of H1 and H2. The profits achieved by H1 and H2 were computed by simulation.

Cases 1-36 were selected to cover a wide range of operating conditions. Cases 1-6, 13-18 and 25-30 represent situations where the value of the mated part gets lower as the “difference” between the halves increases. (For example, different types may correspond to different tolerances, and the halves with the same tolerances may have the best fit.) Cases 7-12, 19-24 and 31-36 represent situations where there are different “qualities” associated with each type, and an assembled component is only worth as much as its lowest quality part.

Tables 1 through 3 show that both heuristics perform well for all 36 cases. H2 performs as well as or better than H1 in all but one of the 36 cases. In fact, the average sub-optimality of H2 was 1.4 % over the 36 test cases. This is encouraging since it is much easier to compute the parameters of H2; the thresholds required to implement H2 were calculated in several minutes for each of the examples. The simulation runs required to determine the thresholds for H1, by comparison, took as long as several hours of CPU time on a Sun Sparcstation 20 machine. The performance of the heuristics declines slightly as the processing rates of the left and right machines become more unequal. Interestingly, H2 was slightly more accurate when the left and right side type probabilities were different than when they were the same.

Finally, we compared the performance of the two heuristics on a typical problem from flat panel display manufacturing. As described in Section 1, in flat panel display manufacturing, “active” and “passive” halves are inspected to determine the location of the defects. The plates are then assembled and cut into smaller products. A typical example has the “passive” and “active” halves cut into four pieces after being mated, to produce 4 displays. In this case, since each half could have defects in four

Case	V_{11}, V_{12}, V_{13} V_{21}, V_{22}, V_{23} V_{31}, V_{32}, V_{33}	l_1, l_2, l_3 r_1, r_2, r_3	h	H1 (% subopt.)	H2 (% subopt.)	Optimal
1	10,7,4 7,10,7 4,7,10	0.33,0.33,0.33 0.33,0.33,0.33	0.05	4.21 (0.9)	4.23 (0.5)	4.25
2	10,7,4 7,10,7 4,7,10	0.33,0.33,0.33 0.33,0.33,0.33	0.02	4.52 (0.0)	4.52 (0.0)	4.52
3	10,7,4 7,10,7 4,7,10	0.50,0.30,0.20 0.50,0.30,0.20	0.05	4.24 (0.5)	4.25 (0.2)	4.26
4	10,7,4 7,10,7 4,7,10	0.50,0.30,0.20 0.50,0.30,0.20	0.02	4.52 (0.2)	4.52 (0.2)	4.53
5	10,7,4 7,10,7 4,7,10	0.60,0.20,0.20 0.20,0.20,0.60	0.05	3.32 (1.2)	3.36 (0.0)	3.36
6	10,7,4 7,10,7 4,7,10	0.60,0.20,0.20 0.20,0.20,0.60	0.02	3.49 (0.9)	3.52 (0.0)	3.52
7	10,6,2 6,6,2 2,2,2	0.33,0.33,0.33 0.33,0.33,0.33	0.05	2.36 (2.1)	2.38 (1.2)	2.41
8	10,6,2 6,6,2 2,2,2	0.33,0.33,0.33 0.33,0.33,0.33	0.02	2.61 (0.4)	2.62 (0.0)	2.62
9	10,6,2 6,6,2 2,2,2	0.50,0.30,0.20 0.50,0.30,0.20	0.05	2.94 (1.7)	2.98 (0.3)	2.99
10	10,6,2 6,6,2 2,2,2	0.50,0.30,0.20 0.50,0.30,0.20	0.02	3.18 (0.9)	3.20 (0.3)	3.21
11	10,6,2 6,6,2 2,2,2	0.60,0.20,0.20 0.20,0.20,0.60	0.05	1.82 (2.7)	1.86 (0.5)	1.87
12	10,6,2 6,6,2 2,2,2	0.60,0.20,0.20 0.20,0.20,0.60	0.02	1.96 (1.5)	1.99 (0.0)	1.99

Table 1: Results for 3-type Cases, $\mu_1 = \mu_2 = 0.50$

Case	V_{11}, V_{12}, V_{13} V_{21}, V_{22}, V_{23} V_{31}, V_{32}, V_{33}	l_1, l_2, l_3 r_1, r_2, r_3	h	H1 (% subopt.)	H2 (% subopt.)	Optimal
13	10,7,4 7,10,7 4,7,10	0.33,0.33,0.33 0.33,0.33,0.33	0.05	3.51 (2.8)	3.51 (2.8)	3.61
14	10,7,4 7,10,7 4,7,10	0.33,0.33,0.33 0.33,0.33,0.33	0.02	3.72 (1.1)	3.70 (1.6)	3.76
15	10,7,4 7,10,7 4,7,10	0.50,0.30,0.20 0.50,0.30,0.20	0.05	3.52 (3.0)	3.53 (2.8)	3.63
16	10,7,4 7,10,7 4,7,10	0.50,0.30,0.20 0.50,0.30,0.20	0.02	3.70 (1.9)	3.70 (1.9)	3.77
17	10,7,4 7,10,7 4,7,10	0.60,0.20,0.20 0.20,0.20,0.60	0.05	2.84 (3.7)	2.93 (0.7)	2.95
18	10,7,4 7,10,7 4,7,10	0.60,0.20,0.20 0.20,0.20,0.60	0.02	2.95 (1.7)	3.00 (0.0)	3.00
19	10,6,2 6,6,2 2,2,2	0.33,0.33,0.33 0.33,0.33,0.33	0.05	1.96 (5.8)	2.00 (3.8)	2.08
20	10,6,2 6,6,2 2,2,2	0.33,0.33,0.33 0.33,0.33,0.33	0.02	2.13 (3.2)	2.15 (2.3)	2.20
21	10,6,2 6,6,2 2,2,2	0.50,0.30,0.20 0.50,0.30,0.20	0.05	2.47 (3.9)	2.49 (3.1)	2.57
22	10,6,2 6,6,2 2,2,2	0.50,0.30,0.20 0.50,0.30,0.20	0.02	2.63 (2.2)	2.64 (1.9)	2.69
23	10,6,2 6,6,2 2,2,2	0.60,0.20,0.20 0.20,0.20,0.60	0.05	1.55 (7.2)	1.64 (1.8)	1.67
24	10,6,2 6,6,2 2,2,2	0.60,0.20,0.20 0.20,0.20,0.60	0.02	1.66 (3.5)	1.71 (0.6)	1.72

Table 2: Results for 3-type Cases, $\mu_1 = 0.60, \mu_2 = 0.40$

Case	V_{11}, V_{12}, V_{13} V_{21}, V_{22}, V_{23} V_{31}, V_{32}, V_{33}	l_1, l_2, l_3 r_1, r_2, r_3	h	H1 (% subopt.)	H2 (% subopt.)	Optimal
25	10,7,4 7,10,7 4,7,10	0.33,0.33,0.33 0.33,0.33,0.33	0.05	2.87 (4.0)	2.92 (2.3)	2.99
26	10,7,4 7,10,7 4,7,10	0.33,0.33,0.33 0.33,0.33,0.33	0.02	3.05 (1.9)	3.07 (1.3)	3.11
27	10,7,4 7,10,7 4,7,10	0.50,0.30,0.20 0.50,0.30,0.20	0.05	2.89 (4.0)	2.93 (2.7)	3.01
28	10,7,4 7,10,7 4,7,10	0.50,0.30,0.20 0.50,0.30,0.20	0.02	3.06 (2.2)	3.07 (1.9)	3.13
29	10,7,4 7,10,7 4,7,10	0.60,0.20,0.20 0.20,0.20,0.60	0.05	2.38 (3.6)	2.46 (0.4)	2.47
30	10,7,4 7,10,7 4,7,10	0.60,0.20,0.20 0.20,0.20,0.60	0.02	2.46 (2.0)	2.50 (0.4)	2.51
31	10,6,2 6,6,2 2,2,2	0.33,0.33,0.33 0.33,0.33,0.33	0.05	1.62 (5.8)	1.66 (3.5)	1.72
32	10,6,2 6,6,2 2,2,2	0.33,0.33,0.33 0.33,0.33,0.33	0.02	1.75 (3.8)	1.78 (2.2)	1.82
33	10,6,2 6,6,2 2,2,2	0.50,0.30,0.20 0.50,0.30,0.20	0.05	2.03 (4.7)	2.08 (2.3)	2.13
34	10,6,2 6,6,2 2,2,2	0.50,0.30,0.20 0.50,0.30,0.20	0.02	2.18 (2.2)	2.19 (1.8)	2.23
35	10,6,2 6,6,2 2,2,2	0.60,0.20,0.20 0.20,0.20,0.60	0.05	1.30 (7.8)	1.38 (2.1)	1.41
36	10,6,2 6,6,2 2,2,2	0.60,0.20,0.20 0.20,0.20,0.60	0.02	1.38 (4.2)	1.43 (0.7)	1.44

Table 3: Results for 3-type Cases, $\mu_1 = 0.66, \mu_2 = 0.33$

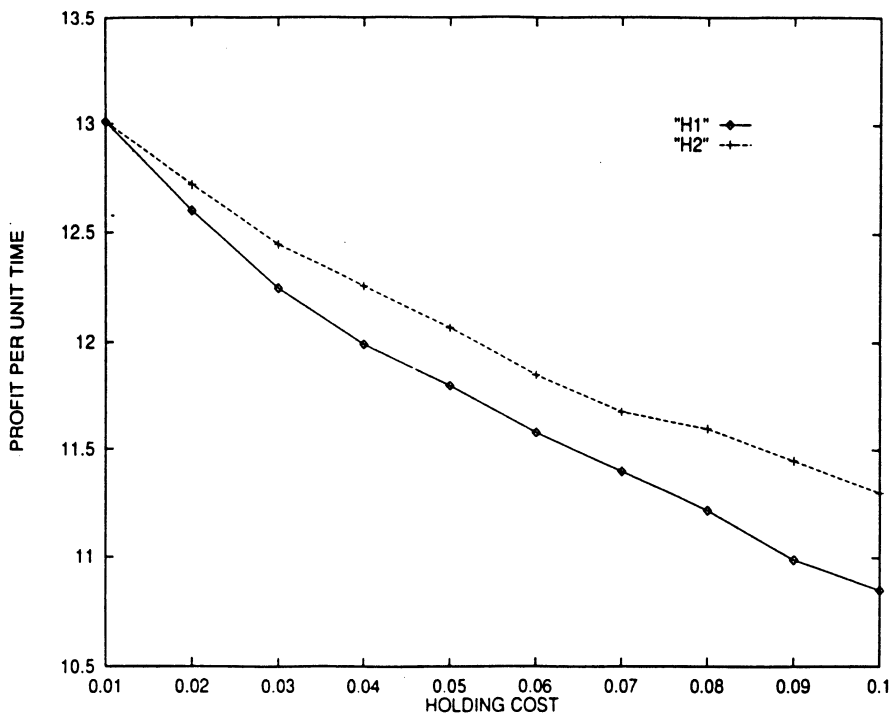


Figure 1: Performance of H1 and H2 for 16-type problem

different locations, there are 16 possible types of halves. Suppose that the probability of defect in any location is 0.3, independent of defects at other locations, and a display is worth \$10 if it has no defects on either of its halves. Therefore mating “passive” and “active” halves that both have no defects is worth \$40. On the other hand, mating an active plate that has a defect in the location of the third display with a passive plate that has defects in the location of the second and fourth displays will result in production of only one 1 display (in the location of the first display) and the total mate is worth only \$10. We assumed that the machines producing the left and right halves had exponential distributions with rate 0.5.

Since this problem has 16 types, it cannot be solved to optimality using dynamic programming. Even if the maximum inventory of a given type were limited to be at most 40, there would be over 10^{28} states in the MDP. Figure 3 shows the performance of H2 and H1 as a function of h . For all of the holding cost values displayed, H2 outperforms H1, although in this case the difference in performance is smaller.

6 Conclusions and Further Research

In this paper, we formulated a dynamic type mating problem that arises in many manufacturing environments including flat panel display manufacturing and presented two effective heuristics. Although both heuristics performed well on the examples considered in this paper, further research is needed to test the effectiveness of these and other heuristics on larger scale problems. Further research should also focus on

more complicated systems than the one we considered here (e.g., subassembly lines instead of machines).

Acknowledgements:

This work was supported in part by Grants No: DMI-9308290 and DMI-9424596 from the National Science Foundation and a grant from the Center for Display Technology and Manufacturing at the University of Michigan.

Bibliography

- [1] Ammar, M.H., 1980. "Modelling and Analysis of Unreliable Manufacturing Assembly Networks with Finite Storage," MIT Laboratory for Information and Decision Sciences, Report, LIDS-TH-1004.
- [2] Bhat, U.N., 1986. "Finite Capacity Assembly-like Queues," *Queueing Systems, Theory and Applications*, **1**, 85-102.
- [3] Bonomi, F. 1987. "An Approximate Analysis for a Class of Assembly-like Queues," *Queueing Systems: Theory and Applications*, **1**, 289-300.
- [4] Duenyas, I, 1994. "Estimating the Throughput of a Cyclic Assembly System," *International Journal of Production Research* **32**, 1403-1419.
- [5] Duenyas, I. and W.J. Hopp, 1992. "CONWIP Assembly with Deterministic Processing and Random Outages," *IIE Transactions*, **24**, 97-111.
- [6] Duenyas, I. and W.J. Hopp, 1993. "Estimating the Throughput of an Exponential CONWIP Assembly System," *Queueing Systems: Theory and Applications* **14**, 135-157.
- [7] Duenyas, I. and M.F. Koblis, 1993. "Release Policies for Assembly Systems," Technical Report 93-15, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI 48109-2117, to appear in *IIE Transactions*.
- [8] Duenyas, I., M.F. Koblis and S.M. Pollock, 1994. "Dynamic Type Mating," Technical Report 94-17, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI 48109-2117.
- [9] Gershwin, S.B., 1991, "Assembly/disassembly systems: an efficient decomposition algorithm for tree structured networks," *IIE Transactions*, **23**, 302-314.
- [10] Hazra, J. and A. Seidmann, 1994, "Performance Evaluation of Closed Tree-Structured Assembly Systems," Technical Report OP93-01, William E. Simon Graduate School of Business Administration, University of Rochester, Rochester, New York, 14627.
- [11] Hopp, W.J., and J.T. Simon, 1989. "Bounds and Heuristics for Assembly-Like Queues," *Queueing Systems: Theory and Applications*, **4**, 137-147.
- [12] Hernandez-Lerma, O. 1989. *Adaptive Markov Control Processes*, Springer-Verlag, New York.
- [13] Iyama, T., Masahiro, M., Goto, S. and Koga, T., 1992. "A Race Matching Method for Ball Bearing Manufacturing", Department of Mechanical Engineering, Iwate University, 4-3-5, Ueda, Morioka 020 Japan.
- [14] Lee, H.S., and S.M. Pollock, 1989. "Approximate analysis for the merge configuration of an open queueing network with blocking," *IIE Transactions*, **21**, 122-129.

- [15] Lippman, S.A., (1975), "Applying a new device in the optimization of exponential queueing systems," *Operations Research*, **23**, 687-710.
- [16] Lipper, E.H., and B. Sengupta, 1986. "Assembly-like queues with finite capacity: bounds, asymptotics and approximations," *Queueing Systems: Theory and Applications* **1**, 67-84.
- [17] Liu, Y.C., and H.G. Perros, 1991. "A decomposition procedure for the analysis of a closed fork/join queueing system," *IEEE Transactions on Computers*, **40**, 365-370.
- [18] Rao, P.C., and R. Suri, 1994, "Approximate Queueing Network Models for Closed Fabrication/Assembly Systems. Part 1: Single Level Systems," *Production and Operations Management*, **3**, No:4, 244-275.
- [19] Saboo, S., and W.E. Wilhelm, 1986. "An approach for modeling small-lot assembly networks," *IIE Transactions*, **18**, 322-334.