An Optimal Design for an Ordered-entry Array Flexible Assembly System

SEYED ALIREZA FEYZBAKHSH*; MASAYUKI MATSUI†**
*Department of Management and Systems
Sharif University of Technology
PO.BOX 11365-8639, Tehran
IRAN

**Department of Systems Engineering, The University of Electro-Communications, Tokyo JAPAN
matsui@se.uec.ac.jp

Abstract: Flexible Assembly Systems (FASs) are finding increasing use in today's industry. An ordered-entry array of unit stations of Generalized CSPSs is designed and studied. The addressed problem is: “given an objective cost function, design a simple FAS to meet the given specifications”.

In the proposed two-stage method, minimum cost of each workstation, which is computed independently by Complex-type method at the first stage, has been used in minimizing the objective function at the second stage. Based on the results from an optimization stand-point, the two-stage method seems to produce reasonable engineering results and is superior to the GA optimization method in computational time.

1 Introduction
The role of Flexible Assembly Systems (FASs) [1] [2] [3] [4] [5] in a computer-integrated manufacturing environment is vital, because of their high production rates, reduction in the cost of assembly, and better quality of the final product. In the planning and design phase of these systems, it is useful to have an appropriate design method that can predict system performance for various operating conditions.

The simple FAS, which is studied here, includes a set of Removal Item Cases (RIC) of Generalized CSPS [6] workstations. Two types of conveyer-serviced production systems were presented, the series type and the ordered-entry type, of which the latter one was recently introduced as a Flexible Assembly System. Here, an ordered-entry array of WSs, forms our FAS. Conveyor-Serviced Production Station (CSPS) [7] is a mathematical model of an unloading station and was generalized by introducing the "design factor" and its operating policy RdSRP. The optimal design problem of a single station of the Generalized CSPS, both fixed and removal item cases were presented in [8], previously. Although there are some works on designing a single station [6] [8] [9] [10] [11], modeling and designing a multi-station as a simple FAS has attracted little attention.

Since both the inter arrival time and service time vary in a probabilistic manner, it is important to design the cycle time of the system, number of WSs, and the buffers. Buffers are divided into two kinds; one is various "look-ahead times" for each WS as time buffers, and the other is the "capacity of in-process inventory" for each WS as a space buffer.

Under these assumptions, the expected production cost, which is the sum of the buffer cost (i.e., buffer cost and in-process inventory under RdSRP) and the delay-and-overflow cost, for each WS, is set up as an evaluation function of the multi-station of Generalized CSPS. Therefore, our objective is to make a decision about the cycle time and each WS’s buffer, in order to minimize the evaluation function.

The difficulties in designing the simple FAS lie in the vast number of decision variables involved and complex interactions among them. In order to find near-optimal solutions, it is necessary to integrate a simulation with optimization techniques. Stochastic optimization based on simulation is sometimes called the Monte Carlo optimization.

Due to the stochastic nature of the problem, we apply the two-stage method as the proposed method used complex-type method (CTM) [8] at the final stage. The cycle time and number of WSs, as the input/output variables are determined at the first stage and then the second stage independently computes buffers (i.e., space and time buffers), using a recently proposed Complex-type method. Furthermore, the proposed methods will be compared with the genetic algorithm (GA) [12] [13] [14] [15], and by some numerical examples it is shown that the proposed two-stage method is superior to the GA in computational time.

In order to use the Complex method [16], some further consideration is needed, because some of the variables (i.e., capacities of the reserves, \( N_i \)) are discrete. Indeed, the range of the discrete variables is not so wide (1<\( N_i <10 \), in this study). Hence, by \( N_i \) times repeating the Complex program over the continuous variables, we can obtain the optimum cost.

GAs and the ideas of representing complicated structures by a simple representation of bit strings, and the power of simple transformations to improve these bit strings, was discussed by Holland [17]. Moreover, the usual form of GA described by Goldberg [18].

In other words, what is going to be attempted is the two-stage method application in a domain where very few techniques are available, i.e. the class of optimization problems where the decision variables are discrete and where the system performance is stochastic.

In the next section of this paper, the problem trying to be designed is introduced, consisting of assumptions and notations, the objective function, and the model of evaluation. The two-stage method, as an optimization technique, is then introduced and proposed, followed by a description of the proposed idea about the independent design of the simple FAS, in the third section. The fourth section includes some numerical considerations of the experiments conducted indicating comparisons between a GA and the two-stage design method. Finally, after a brief conclusion, some tentative considerations and recommendations for future study are offered.

2 A Simple FAS Model

2.1 Definitions and Notations
Many kinds of Flexible Assembly Systems were introduced and discussed recently. In this article an ordered-entry array simple FAS is studied, which is one of the most simple and basic one. It is a distinctive characteristic to have flexibility of routing and management, and it is paid attention to in the design of WSs.

Here we considered the simple FAS, which is introduced by multi-station Removal Item Case (RIC) of Generalized CSPS in a two level structure (see Figure 1). The proposed simple FASs consist of a set of WSs, who are arranged in a predetermined order and transfer mechanism to convey the usables through the WSs. The model of the system considered here is a "no information feedback" simple FAS. Usables are delivered (probably from a central dispatch area) to \( K \) functionally identical WSs by a conveyor. The topology of the simple FAS under consideration is...
shown in figure 1. The stations are arranged along the path of the conveyor in the order 1, 2, ..., K. The WSs may or may not have storage space to store usables received for processing. Usables arrive at the first WS with a mean inter arrival time of \( d \). In addition, usables that cannot be withdrawn from the conveyor by station \( i \) will move to station \( i+1 \) (i.e. the usables is said to overflow from station \( i \) to station \( i+1), i=1, 2, ..., K-1 \); if a usable on the conveyor is not withdrawn by any of the \( K \) stations it will be considered an overflow from the system and stored in an overflow storage area.

![Fig.1 : An Ordered-Entry Simple FAS consisting of some RICs of Generalized CSPS](image)

The time between the last process conducted by the operator and obtaining a new usable from the conveyor is called the "delay time". In other words, the "delay time" is the time that the operator waits for the usable, obtains it from the conveyor, and puts it into the reserve.

The design of an assembly system is a complex task since there are numerous system parameters and variables involved. The problem considered in this paper is formulated as a discrete Monte Carlo optimization problem. The design optimization problem will be addressed as follows: given an objective cost function, design the simple FAS to meet (as closely as possible) these given specifications. The controllable (decision) variables to be optimized in the system are inter arrival time of usables to the system, each of the WS's look-ahead times, and the capacity of the reserve assigned to each WS. The look-ahead time is a positive buffer when the design factor \( \varepsilon \) is:

\[ 0 < \varepsilon < 1 \]

and in this study, only the positive buffer is concentrated on.

A summary of various notations used in this paper is given in table 1.

Table 1: A summary of various notations used in this paper

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K )</td>
<td>Number of WSs</td>
</tr>
<tr>
<td>( S_0 )</td>
<td>Mean of assemble time ( \equiv \bar{x} )</td>
</tr>
<tr>
<td>( T_0 )</td>
<td>Production planning period</td>
</tr>
<tr>
<td>( CE )</td>
<td>Constant establish expense for each WS</td>
</tr>
<tr>
<td>( \sim_i )</td>
<td>( i )th WS where ( i=1,2, ...,K )</td>
</tr>
<tr>
<td>( \varepsilon_i )</td>
<td>Design factor</td>
</tr>
<tr>
<td>( N_i )</td>
<td>Capacity of the reserve</td>
</tr>
<tr>
<td>( C_{ni} )</td>
<td>Look-ahead time ((n=1,2, ...,N))</td>
</tr>
<tr>
<td>( L_i )</td>
<td>Mean number of in-process inventory</td>
</tr>
<tr>
<td>( D_i )</td>
<td>Mean delay per service</td>
</tr>
<tr>
<td>( \eta_i )</td>
<td>Mean number of overflow per service</td>
</tr>
<tr>
<td>( \alpha_i )</td>
<td>Unit cost of the number of in-process inventory for ( T_0 )</td>
</tr>
<tr>
<td>( \beta_{1i} )</td>
<td>Unit cost of each delay's Unit time</td>
</tr>
<tr>
<td>( \beta_{2i} )</td>
<td>Unit cost of each overflow</td>
</tr>
<tr>
<td>( NPr )</td>
<td>Number of production that must be assembled</td>
</tr>
<tr>
<td>( EC_i )</td>
<td>( i )th WSs cost</td>
</tr>
<tr>
<td>( TC )</td>
<td>System total cost</td>
</tr>
</tbody>
</table>

2.2 The Model and Evaluations

The problem which is attempting to be solved consists of finding the optimal system-total-cost of a simple FAS made by multi-station of RIC Generalized CSPS. The expected value of system-total-cost is computed as follows:

\[
\text{Min } \text{System}_{\text{Total Cost}} = F(x)
\]

\[ S.t. \quad x \in X, \quad F(x) = E_w f(x, w) \]

Where \( x \) is a vector of decision variables to be optimized and \( X \) is a set of constraints. In the case being studied in this article, the constraints are the upper and lower bounds for these decision variables. Hence, the presented algorithms are based on the expected value of the system-total-cost rather than its actual value, \( E_w f(x, w) \), which is a random function of \( x \). The \( w \) is a random variable belonging to the appropriate probability space. The randomness comes from the random time between arrival and the random time required to service each usable for different WSs. It is going to be attempted to solve this class of problems using a two-stage design method.

The total variable cost associated with WSs, calculated over some convenient time period, as the objective function for the multi-station of RIC of Generalized CSPS (using the single station production cost discussed in [6] [9] [10] [11] [19] [20]) is as follows:

\[
\text{System}_{\text{Total Cost}} = \sum_{i=1}^{m} \alpha_i L_i + (T_0/(\bar{x} + D_i))((\beta_{1i} D_i) + (\beta_{2i} \eta_i))
\]

The first term on the right hand side is the inprocess inventory cost per each station and the next term is the total delay and overflow cost for each WS (i.e., the delay-and-overflow cost per service multiplied by the

---

2 For our WSs which are made by Generalized CSPS, if the capacity of the storage space set to one it performs just like WSs without any storage space.
number of cycle times in the production planning period).

Therefore, each WS cost function may be written as follows:

\[ EC_i = \alpha_i L_i + Y_i \quad , \quad i=1,2,\ldots,K \]  

(2)

When \( Y_i \) is the term of the delay-overflow cost of \( WS_i \). Under optimum buffer variables \( C_i, N_i \), the cost function of \( WS_i \) is shown by the following formula:

\[ EC_i(d,c_i(d),N_i(d)) = \min_{c_i,N_i} EC_i(d,c_i(d),N_i(d)) \quad \text{for} \quad i=1,2,\ldots,K \]  

(3)

Moreover, the system-total-cost \( TC \) of the minimum cost of whole \( WS \) is as follows:

\[ TC(d,c_i(d),N_i(d)) = \min \sum_{i=1}^{K} EC_i(d,c_i(d),N_i(d)) \]  

(4)

3 Two-stage Design Method

3.1 Independent Design

Since no analytical model is available to solve a complete system design problem, it is necessary to integrate the simulation with an optimization technique in order to study the optimal solution. Due to the stochastic nature of the problem, the two-stage method is applied as the proposed method using CTM at the second stage. The method involves the use of some considerations to calculate the two critical design parameters (e.g., "cycle time" and "number of \( WS \)) at the initial stage. The discrete event simulation and the stochastic optimization method were applied to determine the possible combinations of controllable variables for improved system total cost at the final stage.

EVALUATION I: \[ \sum_{i=1}^{K} \min_{c_i,N_i} EC_i \]

EVALUATION II: \[ \min_{c_i,N_i} TC \]

A comparison was conducted between the sum of minimum costs of each WS found by CTM at the second stage (EVALUATION I) and the minimum system total cost by a GA (EVALUATION II), which found the optimum design of the whole variables at the same time. However, it is theoretically clear that:

\[ \sum_{i=1}^{K} \min_{c_i,N_i} EC_i \geq \min_{c_i,N_i} TC \]  

(5)

The numerical experiences show that the difference between two methods is negligible, but the computational time of EVALUATION I is much shorter.

Table 2: A Comparison of the Optimization

<table>
<thead>
<tr>
<th>Method</th>
<th>(using CTM)</th>
<th>(GA)</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVALUATION I</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>EVALUATION II</td>
<td>0.80</td>
<td>1.41</td>
<td>1.5</td>
</tr>
<tr>
<td>( N_i )</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>( c_i )</td>
<td>1.09</td>
<td>0.80</td>
<td>0.9</td>
</tr>
<tr>
<td>( c_2 )</td>
<td>1.38</td>
<td>1.41</td>
<td>1.5</td>
</tr>
<tr>
<td>( c_3 )</td>
<td>3.20</td>
<td>3.19</td>
<td>3.0</td>
</tr>
<tr>
<td>TC</td>
<td>3,817</td>
<td>3,811</td>
<td>3,807</td>
</tr>
<tr>
<td>Time</td>
<td>01:23</td>
<td>08:10</td>
<td>~</td>
</tr>
<tr>
<td>Error with AC</td>
<td>0.26%</td>
<td>0.11%</td>
<td>___</td>
</tr>
</tbody>
</table>

An example for a single station system can validate the performance of the proposed method. Table 2 indicates a comparison between the two methods. The forth column of the table contains the results of an almost all possible combination search (AC). Based on the results tabulated, it can be concluded that the optimum cost found by the proposed idea is negligible compared to the other optimum Costs. As can be seen, the proposed method is superior to that of EVALUATION II (i.e., GA), while the error is considerably small. As it will be shown, the numerical examples of the next section verify this conclusion.

3.2 The Structure of the Proposed Design Method

The structure of the proposed two-stage design method may be found in the following description. In addition, Figure 3 illustrates the corresponding flowchart.

1st Stage: Setting Up the Input /Output Variables

1. Cycle time

The cycle time is initially calculated. Under the production planning amount, \( NPr \), and the production-planning period, \( T_0 \), the cycle time (Cycle-time) can be calculated using the following formula:
Cycle_time = \frac{T_0}{N_{Pr}} (= d) \tag{6}

2. Number of Workstations
Using the mean assembly time for a usable, S_0, and the cycle time (calculated above), the minimum number of the WS, K, which may contain the plan, is next calculated by the following formula:

\[ K = \left\lceil \frac{S_0}{\text{Cycle time}} \right\rceil \tag{7} \]

2nd Stage: Computation of Optimal Buffers (C_i, N_i)
1. According to the independence optimal design of each WS, i.e., EVALUATION I,
Compute the optimal buffers (capacity of the reserve and look-ahead times) for the first station, under the founded Input/Output variables using complex-type method (C_1, N_1).
2. Repeat the above step computation for the next stations when the inter arrival time is the mean of the inter-overflow time of the previous station ((C_2, N_2), (C_3, N_3), ...).

4 Numerical Considerations
Suppose the following parameter is used as an example:
The Production Planning Amount, N_{Pr} = 32,000 assemblies and the Total Working Time = 20 (days) and 7 (hours a day) Following the abovementioned structure, we have:

1st Stage:
1. Cycle time
\[ \text{Cycle time} = \frac{T_0}{N_{Pr}} = \frac{20 \text{(Days)} \times 24 \text{(Hours)} \times 60 \text{(Minutes)}}{32,000 \text{(Pieces)}} = 0.26 \text{(Minutes/Pieces)} \]

2. Number of Workstations
Utilizing the above calculated, Cycle_time=0.26, and a given mean assemble time, S_0=1.25(Mins.), the minimum number of WSs can be calculated as follows:

\[ K = \left\lceil \frac{S_0}{\text{Cycle time}} \right\rceil = \left\lceil \frac{1.25}{0.26} \right\rceil = 5 \]

Figure 4 shows the change of the total cost under the various cycle times. However, the calculated cycle time is \(d = 0.26\), based on this Figure the most suitable cycle time for the case of \(K = 5\ WSs\), is \(d^{*} = 0.31\).

The effect of the total cost for different numbers of WSs is illustrated in Figure 5. The cycle time for all of the cases is fixed and the system overflow cost coefficient is as follows:

\[ 1.0 \leq \beta_{2k} \leq 3.0, \quad \beta_{2i} = 0 \quad (i=1,2,...,K-1). \]

Figure 5: Relation between Total Cost and Number of WS's; \(K=5, T_0=8,400, \varepsilon=0.8, \alpha=1,000, \beta=1.8, S_0=1.25, d=0.26, \beta_{2i}=0, (i=1, 2, ..., (K-1)), \) and \( \beta_{2k}=1.0, 2.0, 3.0 \)
This Figure may clarify the validity of the decided number of WSs. As can be seen, the optimum number of WSs, is around $K = 5$. According to the Figure, it is happened in $K = 4$ for $\beta_2 = 1$. It may be concluded that with a very small $\beta_2$, the effect of the delay cost becomes more than the effect of overflow cost in the total cost function. In other words, with $K = 4$ a very small delay is present, but with a lot of system overflows. However, $K = 4$ cannot satisfy the amount of assemblies during $T_0$. Furthermore, based on the Figure, when the $K$ exceeds 7 the three curves cover each other. It seems for values of $K$ greater than 7, the main part of the total cost is the delay cost, and there is almost no system overflow. Therefore, under the different system’s overflow unit cost, the same cost result is obtained.

Table 3: Two Stage Method, EVALUATION I, using CTM

<table>
<thead>
<tr>
<th>RUN</th>
<th>Station 5</th>
<th>Total Cost</th>
<th>Computational Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N_5$</td>
<td>$c_{51}$, $c_{52}$, $c_{53}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.57, 1.60, 3.83</td>
<td>32781 00:18:49</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.47, 1.90, 3.23</td>
<td>32543 00:19:28</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.50, 2.08, 3.00</td>
<td>32680 00:20:35</td>
</tr>
</tbody>
</table>

2nd Stage:
Optimal Buffers
Under these circumstances, the optimal buffers may be computed by the CTM. Table 3 indicates the results of three times computations under the following conditions: $K = 5$, $T_0 = 8,400$, $\epsilon = 0.8$, $\alpha = 1,000$, $\beta_1 = 1.8$, $S_0 = 1.25$, $d = 0.26$, $\beta_2 = 0$ ($i = 1, \ldots, 4$) and $\beta_3 = 3.0$.
As can be seen, the optimum cost is around $TC = 32400$ and the computational time was less than 20 minutes on a Pentium-200 PC.

Table 4: Genetic Algorithm, EVALUATION II

<table>
<thead>
<tr>
<th>RUN</th>
<th>Station 5</th>
<th>Total Cost</th>
<th>Computational Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N_5$</td>
<td>$c_{51}$, $c_{52}$, $c_{53}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.62, 1.97, 4.07</td>
<td>32160 01:38:50</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.68, 1.81, 3.16</td>
<td>32097 01:42:35</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>0.50, 2.08, 3.12</td>
<td>32956 02:18:23</td>
</tr>
</tbody>
</table>

Furthermore, the results of the GA program (i.e., EVALUATION II) under the same configuration are shown in Table 4. The ranges of the variables are as follows: the unit step for the discrete variable (capacity of the reserve, $N$) is necessarily 1, and for the other variables is $0.1: 1 \leq N \leq 10$, $0.1 \leq d \leq 5.0$, $0.1 \leq c_i \leq 2.0$, $1.0 \leq c_2 \leq 3.0$ and $2.0 \leq c_3 \leq 5.0$. In order to find a good set of GA control parameters for the current problem, a great deal of experimentation was performed. Finally, the numbers, which were used as GA parameters, were: probability of crossover = 0.6, probability of mutation = 0.003, population size = 30, and a 1-point crossover.

Table 5: A Comparison between the methods

<table>
<thead>
<tr>
<th>RUN</th>
<th>EVALUATION I (using CTM)</th>
<th>EVALUATION I (using GA)</th>
<th>EVALUATION II (using GA)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TC</td>
<td>Time</td>
<td>Error</td>
</tr>
<tr>
<td></td>
<td>32,668</td>
<td>00:19:38</td>
<td>0.33%</td>
</tr>
<tr>
<td></td>
<td>32,560</td>
<td>01:02:00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32,071</td>
<td>01:53:50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.9%</td>
<td>1.5%</td>
<td></td>
</tr>
</tbody>
</table>

Moreover, Table 5 illustrates the average of the results of the three runs for the two-stage using CTM at second stage and the average of the same results for the two-stage using GA at the second stage. In addition, the table makes a comparison between these methods and a GA optimization method, which the Input/Output variables and the buffers designed at the same time. As can be seen, the proposed two-stage method using CTM is superior to the other methods in computational time.

Hence, it can be concluded that the proposed method produces accurate optimum variables with minimal computational efforts in comparison with the GA.

5 Conclusion and future study
In this article, a simple FAS using an ordered-entry array of multi-station RIC of Generalized CSPS has been studied. In a practical sense, the main contribution of this paper is a novel approach for the design of the variables, which include Input/Output variables (i.e., cycle time and number of $WS_n$) and buffers (i.e., capacity of in-process inventory for each $WS_n$ and various look-ahead times of $WS_n$). Moreover, a methodology to determine the optimal assemble cost, as the objective function, was developed.

A procedure combining the use of some analytical considerations, discrete event simulation, and heuristic Monte Carlo optimization methods appears to perform well in obtaining optimal/near optimal design configurations for the simple FAS. A two-stage method with application to the simple FAS design optimization problem under study has been proposed and introduced. After some analytical considerations at the first stage, the CTM has been implemented as the optimization method at the second stage. At this stage, the system-total-cost has been computed by finding the cost of each $WS$ Independently.
The proposed two-stage method has been compared to a GA. Through some numerical examples, it was shown that the proposed method is superior to the GA. The computational requirements for a GA run were quite large. Moreover, in the examples provided in the previous section, although there are no significant differences between the results (i.e., estimates of minimum total-system-cost) obtained by these two algorithms, we can find that, in comparison to the GA, the computation time decreased significantly by using the two-stage method (approximately 5 times).

One of the future research areas that may be investigated is the study of more complex FASs using RIC of Generalized CSPS, which involve different types of production assemblies using different service operational stations and variable processing paths.

References: