Proceedings of the 1988 Winter Simulation Conference
M. Abrams, P. Haigh, and J. Comfort (eds.)

Integrating a modified simulated annealing algorithm with the simulation of a manufacturing system to optimize buffer sizes in automatic assembly systems

Akif Asil Bulgak and Jerry L. Sanders
Department of Industrial Engineering
University of Wisconsin-Madison
1513 University Ave.
Madison, WI 53706

ABSTRACT

Selecting appropriate buffer sizes for the transport systems of automated manufacturing systems is a complex task that must account for random fluctuations in production rates by individual machines as well as the transport delays that are a part of the material handling system. If the buffer sizes are too large then the transport delays are excessive and more in-process inventories must be loaded into the system to accomodate the large buffer sizes. If the buffer sizes are too small then small processing delays will cause the buffers to fill and upstream workstations will be blocked from releasing completed workpieces. This paper integrates an extension of the simulated annealing algorithm used in combinatorial optimization with a discrete event simulation of the manufacturing system to find optimal buffer sizes for asynchronous assembly systems which involve automated inspection as well as automated assembly.

I. INTRODUCTION

I.1. Automatic Assembly Systems

An Automatic Assembly System (AAS) is an array of workstations, and a transfer mechanism to convey the assemblies from one workstation to the next one. The assemblies are transported by *work carriers* or *pallets* that hold the assemblies. The workstations usually consist of automatic *part-feeders* and *workheads* to perform assembly operations. In most of the AASs, the workheads remain stationary and the workcarriers are transferred intermittently (Boothroyd et al. 1982). The workstations are generally arranged in a closed circular fashion, and a fixed number of pallets perpetually circulate in the system to link the workstations. Simple insertion, nut and screw running, and welding are some examples of typical assembly operations.

According to the transfer systems used, AASs can be categorized as *synchronous* and *asynchronous* systems. In the former category, the transfer of all assemblies occurs at fixed time intervals and the whole system is paced by the speed of the transport mechanism. On the other hand, in the latter category, the assembly is transported on a pallet which is moved by a transfer chain/conveyer. In asynchronous systems pallets move independently. As a result of this, to a certain extent, workstations can operate independently.

Some AASs have *test (inspection) stations*, and *repair loops*. At the test stations, assemblies are tested according to a sampling plan, and those which are found to be nonconforming (e.g. those with improperly assembled components) are diverted into a secondary loop where the defects are corrected at a repair station and the workpiece is again allowed into the main loop. Throughout this paper, we shall be discussing about the asynchronous automatic assembly systems that have test stations and repair loops.

I.2. Parameters that Affect the Performance of an AAS

Performance of an AAS depends on various parameters. These parameters should be considered at the design stage. The major parameters are the cycle times of the workstations, the rate of occurence of parts jams for each workstation, the distribution of random amount of time to clear jams, the buffer sizes between the pairs of workstations, the unit transport time for the pallets, the total number of pallets on the line, the average fraction of defective parts produced, and the sampling policy for the quality control of the finished assemblies (however, different quality control policies can be employed at the various stages of the production). In general, the cycle times of workstations are deterministic. But jams or misassembled parts, due to their random clear times, introduce a randomness to the cycle times (see section I.4). The jam occurrences, which make the analysis of an AAS more difficult, represent the first key factor reducing the line efficiency. In an AAS, the parts jams may occur whenever a defective part is assembled or there is a bad positioning of the part being assembled. An other example of the parts jam can be given for a robotic assembly station when the robot drops the part accidently and moves back to the original position to pick a new part (Liu and Sanders 1986). The second key factor affecting the line efficiency is the *starvation* and the *blocking* effects. If the preceding buffer of a particular workstation is empty, the workstation is starved and if the following buffer of the workstation is full, then it is blocked. In both cases, the workstation cannot operate. Starvation and blocking are often caused by the insufficient buffer sizes between the stations, and the total number of pallets in the system. Mismatches in the sampling time requirements required for reasonably complete system tests compared to production cycle times constitute the third key factor which reduces the line efficiency. Due to the very high production rates of AASs, the problem of congestion caused by the final test station must be faced. This congestion could be overcome again by the appropriate allocation of buffers, and employing looser sampling plans. Looser sampling plans reduce the *Average Outgoing Quality (AOQ)* of the product where AOQ is defined as percent defective product going to inventory. A clear tradeoff exists between AOQ and the production rate (Sanders 1985, Bulgak and Sanders 1987a, Bulgak and Sanders 1987b).

I.3. Test and Repair Stations, Inspection Policies, the Effect of the Counter-Flow Loop

The assemblies are circulated through the workstations in the main loop and finally enter the test station where they are inspected according to a sampling policy. After the inspection, the assemblies split into two groups. While the non-defective assemblies go to the inventory, those which are identified to be defective are diverted into the counter-flow loop for repair. Naturally, the cycle times of the repair stations are relatively much bigger than those of the workstations on the main loop. Although, in general, the incoming fraction defective (p') coming to the test station is small, a bottleneck may occur due to the bigger repair times. The repaired assemblies are then taken back to the main loop where they merge with the assemblies in the main loop for inspection. As is the case in many AASs, in our examples the workstation right before the inspection station in the main loop handles the merging phenomenon. Since this station is fed by two sources, the buffer sizes preceding, and succeeeding the station become more crucial.

Depending on our objectives and policies, we can employ various sampling plans in the test station. These plans are the ones for continuous production. As well as 100% inspection, the sampling plans suggested by Dodge (1943), Wald and Wolfowitz (1945), and Girschick (Duncan 1974) are good examples of rectifying inspection plans for continuous production. Because of their wide applications in industry, we chose to use Dodge CSP-1 plans in our examples. Obviously, 100% inspection yields the best AOQ while reducing the line efficiency of the AAS. Dodge CSP-1 plans, on the contrary, cannot guarantee an equally good AOQ but improve the line efficiency.

At the start of the CSP-1 plan all product is inspected 100 percent. As soon as i consecutive units of product are found to be free of defects, 100 percent inspection is discontinued and only a fraction f of the units are inspected. These individual sample units are to be selected one at a time at random from the flow of the product. If a sample unit is found defective, reversion is immediately made to 100 percent inspection and cycle is completed (Duncan 1974). The three important parameters of a Dodge CSP-1 plan are f, i, and *Average Outgoing Quality Limit (AOQL)*. AOQL is defined as the maximum possible value of AOQ which cannot be exceeded no matter what the value of p' is. There may be associated with each CSP-1 plan an overall AOQL, the specific value depending on the values of i and f combinations.

I.4. The AAS Model

In this study, an AAS with M workstations in the main loop and 1 repair station in the counter-flow loop is discussed as an example. Figure 1 shows a scheme of the AAS under study. The workstation number 1 is the load/unload station. The workstation number M-2 is the one which handles the merging of the assemblies coming from main and repair loops. Station number M-1 is the inspection and test station, and number M is the repair station. All other workstations perform various assembly operations which are immaterial in this study (Figure 1).

As examples, either 6 station or 10 station AASs are used in the buffer size optimization problem. Some stations in these AASs are subject to jam with a 5% jam rate. Taking the random jam occurences into consideration, we model the station cycle times (T) in terms of two components as follows.

$$T = D + XR$$

where D is the deterministic cycle time and XR is the random component due to the random jam occurences. In the expression XR, R is the time required to clear a jam, and X is an indicator variable that takes the value of 1 if a jam occurs, and 0 otherwise. In some examples R is assumed to be a deterministic quantity of 15 time units, and in some others, it is assumed to be a random variable with geometric distribution having a mean of 15 time units. Cycle time for station number M-2 (where merging occurs) is 1 time unit. Cycle times for test and repair stations are 7 and 20 time units respectively. All other workstations have the deterministic components (D) of cycle times of 5 time units. The fraction defective p' is considered to be 5% in all the runs. Buffer sizes in the system are numbered according to the numbers of workstations preceding them. Since workstation M-2 is fed by two sources, and therefore having two buffers that precede it, these buffers are called b(M-2), and b(M-2)'. b(M-2)' is the buffer preceding station M-2 on the repair loop (Figure 1).

I.5. Buffer Size Allocation in AASs as a Combinatorial Problem

As noted above, in asynchronous automatic assembly systems the assembly is transported on a pallet where the pallets ride or float on a transfer chain/conveyer. This transfer mechanism is called a free-transfer mechanism. The free-transfer mechanisms permit pallets to move independently and to be queued in front of a workstation. The buffer size determines the number of pallets that can be queued in front of a particular workstation. In other words, it is the spacing between the pairs of workstations where a finite number of pallets can fit. To a certain extent, an autonomy from workstation to workstation could be achieved in an AAS due to the buffers. In many cases, this autonomy is important in the context of line efficiency. Reductions in line efficiency caused by the random jam occurrences, and the blocking and starvation effects can largely be avoided or considerably be reduced by the careful allocation of buffers.

Practically, the possible buffer configurations of an AAS are finite. If N is the total number of pallets that perpetually circulate through the AAS, then any buffer size in a real world system should be in the interval [1,N]. Provided that we have M buffer spacings between each pair of workstations, then N^M different buffer configurations are possible. N^M, which constitutes the size of the configuration space, is a big number even for small AASs. For example, a 10 station AAS with 11 buffer spacings in Figure 1, N^M is $4.1943*10^{17}$ for a total of 40 pallets in the system, and $3.62797*10^{19}$ for a total of 60 pallets in the system. In general, industrial AASs have more workstations and hence buffer spacings and more total number of pallets than that in our example. The problem is to find the buffer configuration which yields the best performance value (e.g. throughput) for a given AAS. Hence, we view the buffer size allocation problem as a combinatorial, discrete, and multivariate Monte Carlo optimization problem.

II. PREVIOUS WORK

Optimization by simulated annealing was independently presented by Kirkpatrick et al. (1983) and Cerny (1985). It was successfully implemented for various deterministic optimization problems and some theoretical studies regarding its convergence characteristics have been done. This section contains a brief review in the area.

Kirkpatrick, Gelatt, and Vecchi (1983) introduced optimization by simulated annealing. They describe a detailed analogy between annealing in solids and multivariate combinatorial optimization that provides a framework for optimization of the properties of large and complex systems. They discuss the Metropolis procedure (Metropolis et al. 1953) in statistical mechanics which is the analog of the optimization method they propose. They also discuss implementation of the algorithm. They provide examples of various optimization problems that are all very complex and difficult to solve due to their sizes. Those problems include physical design of computers, and the travelling salesman problem. They conclude that simulated annealing may be a better model when compared to iterative improvement in combinatorial optimization. Vecchi and Kirkpatrick (1983) applied simulated annealing to global wire routing for both idealized and actual designs of realistic size and complexity. They conclude that simulated annealing results are better than those obtained by the conventional heuristic methods. Hence, they use them as a standard against which to compare several sequential or greedy strategies commonly employed in automatic wiring programs. Cerny (1985) presented a Monte Carlo algorithm to find approximate solutions of the travelling salesman problem. The algorithm generates randomly the permutations of the stations of the travelling salesman trip, with probability depending on the length of the corresponding route. Reasoning by the analogy with statistical thermodynamics, he uses the probability given by the Boltzmann-Gibbs distribution. He states that by using this

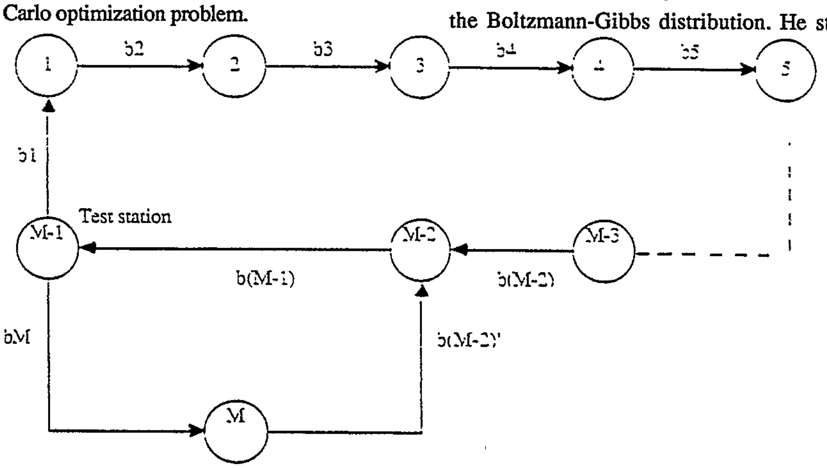

Figure 1. Scheme of an assembly system with a counter-flow repair loop

simple algorithm, it is possible to get very close to the optimal solution or find the true optimum. He demonstrates several examples. He conjectures that his thermodynamical approach to travelling salesman problem can offer a new insight into optimization problems and can suggest efficient algorithms for solving them.

Hajek (1988) establishes cooling schedules for optimal annealing. He states that simulated annealing is a randomized algorithm where the level of randomization is determined by a control parameter T, called temperature. T converges to zero according to a deterministic "cooling schedule". In this theoretical study, he gives a simple necessary and sufficient condition on the cooling schedule for the algorithm state to converge in probability to the set of globally minimum cost configurations. Sasaki and Hajek (1986) used simulated annealing for the problem of finding a maximum cardinality matching in a graph. Mitra, Romeo, and Sangiovanni-Vincentelly (1986) developed a theoretical analysis of simulated annealing based on its precise model, a time-inhomogeneous Markov chain. They analyzed the finite-time behaviour of simulated annealing and obtained a bound on the departure of the probability distribution of the state at finite time from the optimum. This bound gives an estimate of the rate of convergence and insights into the conditions on the annealing schedule which gives optimum performance. Wilhelm and Ward (1987) present an application of the simulated annealing method to solve the quadratic assignment problem. They test the performance on a set of standard and newly created problems and compare the results of simulated annealing with the results from other known heuristics. They conclude that simulated annealing can yield better solutions at comparable CPU times under certain conditions.

III. A MODIFIED SIMULATED ANNEALING ALGORITHM

Simulated annealing is a randomized algorithm which has been developed for the deterministic multivariate combinatorial optimization problems. In this paper, we present a modified version of this algorithm in an attempt to extend its application to the domain of Monte Carlo optimization. Simulated annealing has proved to be a useful and reasonably general purpose tool for (deterministic) combinatorial problems. As with many other general purpose tools, its efficiency is not always as great as one might hope. However it does provide a general purpose approach for a wide class of difficult problems. Unfortunately when one extends the problem class to stochastic combinatorial problems where the value of the objective functional can only be estimated, there are few methods which offer one any hope of solving the complex real-world optimization problems that arise. The objective of this paper is to integrate an extension of the simulated annealing algorithm used in combinatorial optimization with discrete event simulation of the manufacturing system to find optimal buffer sizes for asynchronous assembly systems which involve automated inspection and repair as well as automated assembly.

The basic idea in the modified version of simulated annealing is to make the comparisons based on whether or not the values of the objective functionals indicate statistically significant differences at each iteration. The differences between the values of objective functionals are decided to be statistically significant or not based on the confidence intervals set for these values. Each confidence interval is set based on a minimum of two independent replications of the discrete event simulation. In a comparison of the current value of the objective functional with the previous one, if the current value is not smaller and yet not significantly greather than the previous one, one more simulation replication is conducted, the confidence interval is updated, and another comparison is made. This process is repeated until these two values of the objective functionals are found to have a statistically significant difference or a maximum allowed number of replications is achieved. If they have a significant difference then the proposed configuration is accepted. Otherwise, the iteration is ignored, the temperature is not updated, and a new iteration is started based on the previous configuration with the current temperature. If the value of the objective functional of the proposed configuration is smaller (not necessarily significantly smaller) than the current one, this configuration is accepted with a certain probability which is a function of the values of the objective functionals compared and the current temperature.

In section I we made a brief introduction to automatic assembly systems, and described the discrete event simulation model used as well as the buffer size allocation as a combinatorial problem, and in section II we reviewed some of the previous studies in simulated annealing. The rest of the paper is organized as follows. In section III, the new version of the simulated annealing algorithm is proposed which permits its extension to Monte Carlo combinatorial optimization problems. In section V the applications of the algorithm to the problem stated above will be demonstrated.

III.1. Problem Formulation

We formulate the problem as follows:

$$\max : \text{obj } F(x) \,;\; x \text{ in } X$$
$$F(x) = E_{\omega} f(x, \omega)$$

where x is the vector of decision variables, X is a set of constraints, and ω is a random variable belonging to the appropriate probability space.

III.2. Notation

In the algorithm and implementation the following additional notation will be used:

j : iteration number
x_j : the vector of decision variables at iteration j
x^* : the current best vector of decision variables
$N(x^*)$: the set of all possible neighboring configurations of x^*
$C(x_j)$: the cardinality of $N(x_j)$
K : the number of decision variables
$F(x_j)$: the objective functional whose value is estimated by discrete event simulation at x_j
r : number of discrete event simulation replications currently performed
R : maximum number of replications allowed where R is chosen *a priori*
$m_{x_j r}$: statistical estimate of the mean of $F(x_j)$ based on r replications
$v_{x_j r}$: statistical estimate of the variance of $F(x_j)$ based on r replications
$CI_{x_j r}$: a confidence interval defined as $[\, m_{x_j r} - t_{\alpha/2,\, r-1} (v_{x_j r})^{1/2}, m_{x_j r} + t_{\alpha/2,\, r-1} (v_{x_j r})^{1/2} \,]$
u : a random variable uniformly generated in the interval [0, 1]
c : a constant which is greater or equal to the height of the highest local maximum which is not a global maximum state
T_j : the value of temperature at iteration j.

III.3. The Algorithm

For a cooling schedule $T_j=T(j)$ that has a parametric form as a function of j, the algorithm can be stated in terms of the following steps:

Step 1. Determine x_j arbitrarily at $j=0$.

Step 2. Assign $x_j \rightarrow x^*$ and compute $m_{x^* jr}$ where $r=R$ at $j=0$.

Step 3. Generate another x_j configuration randomly where $x_j \in N(x^*)$.

Step 4. Compute $m_{x_j r}$ and $CI_{x_j r}$ based on r replications where initially $r=2$ at j.

Step 5a. Check whether the interval $[\, m_{x_j r}, m_{x_j r} + t_{\alpha/2,\, r-1} (v_{x_j r})^{1/2}]$ includes $m_{x^* jr}$. If it does, go to Step 5b. If it does not, go to Step 6a.

Step 5b. Set $r \rightarrow r+1$. If $r < R$, go to Step 4. If $r=R$, go to Step 3.

Step 6a. If $m_{x_j r} > m_{x^* jr}$, go to Step 6b. If $m_{x_j r} < m_{x^* jr}$, go to Step 6c.

Step 6b. Accept x_j configuration. Assign $x_j \rightarrow x^*$. Go to Step 7.

Step 6c. Generate u. If $u < \exp\{(m_{x_j r} - m_{x^* jr}) / T_j\}$, then accept x_j configuration, and assign $x_j \rightarrow x^*$. Otherwise, reject x_j. In both cases, go to Step 7.

Step 7. Set $j \rightarrow j+1$, and update T_j (see section IV). Check the stopping criterion (see section IV). If it is satisfied, go to Step 8. Otherwise, go to Step 3.

Step 8. Stop.

IV. IMPLEMENTATION AND NUMERICAL EXAMPLES

The buffer spacing allocation in asynchronous assembly systems is approached as a combinatorial problem and the annealing algorithm described in the previous section is implemented. The AAS under study was described in section I. The objective function to be maximized is the number of assemblies produced by the last workstation of AAS per unit time. Any proposed x_j configuration is generated from the set of neighboring configurations of x^* that is $(N(x^*))$. If $b_{x^*,k}$ is an individual buffer size in x^* where $k=1, 2, \ldots, K$, and K is the number of decision variables (buffer spacings) in the system, then the set $N(x^*)$ consists of all possible combinations of $b_{x^*,k}+1$, $b_{x^*,k}+2$, $b_{x^*,k}-1$, $b_{x^*,k}-2$ for all k's. The probability of generating an x_j configuration for a given x_j^* configuration is simply $1/4^K$. However, each buffer spacing generated is constrained by the interval [1,N/3]. Even though in section I.5, we have stated that all buffer sizes in a real world AAS should be in [1,N], our past experience on AAS shows that in most cases, in optimal configurations, the buffer sizes do not exceed N/3. Based on this heuristic, narrowing the interval to [1,N/3] enables us to save redundant iterations and avoid generating non-optimal configurations.

In our runs, we used a cooling schedule such that a single comparison is made at a certain temperature after which the temperature is reduced according to a certain function for the next comparison. The function that we use for updating the temperature has the parametric form of $T_j=c/\log(1+j)$ is the one proposed by Hajek (1988). Hajek defines c as a constant which is greater or equal to the depth of the deepest local minimum which is not a global minimum configuration. If c is greater or equal to the depth mentioned above, the necessary and sufficient condition on the cooling schedule for the algorithm state to converge in probability to the set of globally minimum (maximum) configurations is satisfied. Proofs of convergence for deterministic optimization cases are provided in (Hajek 1985). Even though the original convergence proofs do not apply to applications of simulated annealing for the optimization of stochastic systems, our experimental results indicate that this cooling schedule gives us better results as opposed to others such as $T_j = a^j T_0$ where $0<a<1$. In the cooling schedule $Tj=c/log(1+j)$, the value of c is determined empirically.

In stochastic optimization problems, stopping criteria are generally difficult to determine. In practice, one generally decides to stop either when there is no more or very small improvements in the values of the objective functional or when a maximum number of iterations is achieved. We decided to stop after reaching a maximum number of iterations, hence a very low temperature where we can regard that the system is "frozen".

Discrete event simulation is used to observe the performance of the AAS described in detail in the previous section for the given buffer configuration. The following tables show a summary of our simulated annealing runs and results. A single comparison is performed at a certain temperature for each of the runs. R (the maximum number of replications allowed at each iteration) is chosen *a priori* to be 10. The iteration numbers do not include the cases where the interval $[\, m_{xjr}, m_{xjr} + t_{\alpha/2,\, r-1} (v_{xjr})^{1/2} \,]$ includes m_{x^*jr}. The elements of x^* vector are the buffer sizes denoted by $b_1, b_2, b_3, b_4, b_5, b_6, b_{5'}$ for 6 station AAS examples, and by $b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{8'}$ for 10 station AAS examples respectively (figure 1).

Table 1. A 6 station AAS with 24 pallets where stations 1, 2, and 3 are subject to jam. Time required to clear a jam is deterministic. A Dodge CSP-1 plan with AOQL: 6.5%, f: 0.20, i: 10 is employed. c: 0.001

iteration	x^*	m_{x^*jr}	r	80% confidence intervals
1	(3,3,3,3,3,1,1)	0.13071	10	
2	(1,4,4,1,5,1,2)	0.13194	2	[0.1313, 0.1326]
3	(3,2,6,2,7,3,1)	0.13372	2	[0.1332, 0.1342]
4	(5,1,8,1,8,5,2)	0.13326	2	[0.1323, 0.1342]
5	(6,2,7,2,7,7,3)	0.13543	2	[0.1349, 0.1359]
25	(5,5,5,4,7,5,1)	0.13874	3	[0.1365, 0.1410]
30	(6,7,7,5,8,4,2)	0.14102	3	[0.1395, 0.1426]
50	(8,8,6,6,8,7,4)	0.14186	3	[0.1406, 0.1431]
99	(8,7,7,4,8,6,5)	0.14163	3	[0.1402, 0.1430]
380	(7,8,5,5,8,5,7)	0.14068	2	[0.1399, 0.1415]
388	(8,8,8,5,8,8,7)	0.14186	3	[0.1406, 0.1431]
399	(8,8,8,5,8,8,7)	0.14186	3	[0.1406, 0.1431]

Table 2. A 6 station AAS with 24 pallets where stations 1, 2, and 3 are subject to jam. Time required to clear a jam is a random variable with a geometric distribution. A Dodge CSP-1 plan with AOQL: 6.5%, f: 0.20, i: 10 is employed. c: 0.001

iteration	x^*	m_{x^*jr}	r	80% confidence intervals
1	(3,3,3,3,3,1,1)	0.12557	10	
2	(1,4,4,1,5,1,2)	0.12898	2	[0.1277, 0.1302]
3	(3,2,6,2,7,3,1)	0.13064	3	[0.1271, 0.1342]
5	(4,3,8,3,6,5,2)	0.13061	3	[0.1294, 0.1318]
10	(5,4,8,1,5,4,4)	0.13391	2	[0.1288, 0.1390]
20	(4,6,7,2,7,6,2)	0.13348	4	[0.1302, 0.1367]
25	(6,4,8,1,5,4,4)	0.13344	4	[0.1300, 0.1369]
50	(8,8,6,2,8,4,8)	0.13971	2	[0.1328, 0.1466]
417	(8,8,6,2,8,4,8)	0.13971	2	[0.1328, 0.1466]
499	(8,8,6,2,8,4,8)	0.13971	2	[0.1328, 0.1466]

Table 3. A 6 station AAS with 24 pallets where stations 1, 2, and 3 are subject to jam. Time required to clear a jam is deterministic. A Dodge CSP-1 plan with AOQL: 6.5%, f: 0.066, i: 19 is employed. c: 0.001

iteration	x^*	m_{x^*jr}	r	80% confidence intervals
1	(3,3,3,3,3,1,1)	0.13298	10	
6	(1,4,4,1,5,1,2)	0.13278	2	[0.1267, 0.1388]
10	(1,4,4,1,5,1,2)	0.13278	2	[0.1267, 0.1388]
20	(1,5,5,2,4,2,1)	0.13277	5	[0.1298, 0.1358]
27	(4,4,8,5,4,1,1)	0.13699	2	[0.1307, 0.1427]
30	(3,6,7,4,5,3,3)	0.13656	6	[0.1344, 0.1387]
34	(5,7,5,5,6,2,5)	0.13907	2	[0.1329, 0.1452]
40	(6,8,4,8,8,1,4)	0.14151	3	[0.1381, 0.1449]
100	(6,8,4,8,8,1,4)	0.14151	3	[0.1381, 0.1449]
399	(6,8,4,8,8,1,4)	0.14151	3	[0.1381, 0.1449]

Table 4. A 10 station AAS with 40 pallets where stations 2, 4, 6 and 7 are subject to jam. Time required to clear a jam is a random variable with a geometric distribution. A Dodge CSP-1 plan with AOQL: 6.5%, f: 0.20, i: 10 is employed. c: 0.001

iteration	x^*	m_{x^*jr}	r	80% confidence intervals
1	(3,3,3,3,3,3,3,3,3,1,1)	0.12146	10	
6	(4,4,5,4,2,5,4,2,4,2,3)	0.12506	2	[0.1196, 0.1305]
27	(5,6,7,5,4,4,5,1,5,3,2)	0.13187	2	[0.1213, 0.1425]
43	(7,7,9,6,3,3,4,1,3,5,3)	0.13184	6	[0.1283, 0.1354]
45	(9,9,10,7,1,2,5,3,5,7,4)	0.13176	6	[0.1283, 0.1352]
46	(8,8,8,8,2,3,7,4,3,5,5)	0.13692	3	[0.1322, 0.1417]
116	(9,7,9,9,3,5,6,6,5,3,4)	0.13684	3	[0.1319, 0.1418]
128	(10,9,10,8,1,7,5,5,6,4,5)	0.13656	6	[0.1331, 0.1400]
132	(13,5,11,11,3,3,9,9,7,3,5)	0.13933	4	[0.1356, 0.1431]
499	(13,5,11,11,3,3,9,9,7,3,5)	0.13933	4	[0.1356, 0.1431]

V. CONCLUSIONS

In this study we have made an attempt to extend the application of simulated annealing to the domain of Monte Carlo optimization. As an example, we discussed buffer size allocation problem in asynchronous assembly systems, and presented some preliminary experimental results. Simulated annealing is known to yield "high quality" solutions to deterministic optimization problems. We conjecture that simulated annealing has the potential in providing "good" solutions of not only deterministic but also stochastic combinatorial optimization problems. The convergence characteristics of simulated annealing in optimization of stochastic systems is the subject of our ongoing research.

REFERENCES

Boothroyd, G., Poli, C. and Murch L. E. (1982). *Automatic Assembly.* Marchel Dekker.

Bulgak, A. A., and Sanders J. L. (1987a). Combined stochastic optimization of production rate and product quality control in automatic assembly systems. In: *Proceedings of the 1987 ASME International Computers in Engineering Conference* (R. Raghavan, and T. J. Cokonis, eds.) New York, New York, 2, 99-103.

Bulgak, A. A., and Sanders J. L. (1987b). Stochastic design optimization of åsynchronous assembly systems with inspection stations and counter-flow repair loops. Technical Report No. 87-10, Department of Industrial Engineering, University of Wisconsin-Madison.

Cerny, A. (1985). A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications* 45., 41-51.

Dodge, H. F. (1943). A sampling inspection plan for continuous production. *Annals of Mathematical Statistics* 14, 264-279.

Duncan, A. J. (1974). *Quality Control and Industrial Statistics*, Fourth Edition. Irwin, Ill.

Hajek, B. (1988). Cooling schedules for optimal annealing," *Mathematics of Operations Research* 13 (2), 311-329.

Kirkpatrick, S., Gelatt Jr. C. D., and Vecchi M. P. (1983). Optimization by simulated annealing. *Science* 220, 671-680.

Liu, C. M., and Sanders J. L. (1986). Stochastic design optimization of flexible assembly systems. *Proceedings of the second ORSA/TIMS Conference.*

Metropolis, N., Rosenbluth A., Rosenbluth M., Teller A., and Teller E. (1953). Equation of state calculations by fast computing machines," *Journal of Chemical Physics* 21, 1087-1092.

Mitra, D., Romeo F., and Sangiovanni-Vincentelly A. (1986). Convergence and finite-time behavior of simulated annealing. *Advanced Applied Probability* 18, 747-771.

Sanders, J. L. (1985). Statistical quality control for high speed automatic manufacturing systems. *Proceedings of the 1985 ASME International Computers in Engineering Conference*, Boston, MA, 2, 433-438.

Sasaki, G. H., and Hajek B. (1986). The time complexity of maximum matching by simulated annealing. Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign.

Vecchi, M. P., and Kirkpatrick S. (1983). Global wiring by simulated annealing," *IEEE Transactions on Computer Aided Design*, Vol. CAD-2, No. 4., 215-222.

Wald, A. and Wolfowitz, J. (1945). Sampling inspection plans for continuous production which insure a prescribed limit on the outgoing quality. *Annals of Mathematical Statistics* 16, 30-49.

Wilhelm, M. R., and Ward T. L. (1987). Solving quadratic assignment problems by "Simulated Annealing", *IIE Transactions* 19 (1), 107-119.

AUTHORS' BIOGRAPHIES

Akif Asil Bulgak is a PhD student in the Department of Industrial Engineering at the University of Wisconsin-Madison. He received his BS in Mechanical-Industrial Engineering from the Technical University of Istanbul, Turkey, in 1982 and MS in Industrial Engineering from the University of Wisconsin-Madison in 1985. His research interests are stochastic modeling of manufacturing systems, stochastic optimization, design and analysis of automatic assembly systems.

Jerry L. Sanders is a Professor of Industrial Engineering and Professor of Preventive Medicine at the University of Wisconsin-Madison. He was chairman of the Department of Industrial Engineering from 1975 to 1978. From 1963 to 1974 he was a Professor of Systems and Industrial Engineering at the University of Arizona. He is the author of several research papers in manufacturing systems, operations research, and health care systems analysis.

Akif A. Bulgak
Department of Industrial Engineering
University of Wisconsin-Madison
1513 University Avenue
Madison, WI 53706 U.S.A.
Phone: (608) 263 - 7784

Jerry L. Sanders
Department of Industrial Engineering
University of Wisconsin-Madison
1513 University Avenue
Madison, WI 53706 U.S.A.
Phone: (608) 263 - 5784