TECHNICAL RESEARCH REPORT

Institute for Systems Research

Reactive Operations Scheduling for Flexible Manufacturing Systems

by J.S. Dhingra, K.L. Musser and G.L. Blankenship

The Institute for Systems Research is supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

TR 93-21

Reactive Operations Scheduling for Flexible Manufacturing Systems*

J. S. Dhingra, K. L. Musser and G. L. Blankenship

Abstract

In this paper we present a reactive operations scheduling methodology for online control of manufacturing systems. We present description of the Texas Instruments electronic circuit board assembly unit at Johnson City, TN, which was used as a model for the development of the reactive operations scheduling scheme. Based on the manufacturing system model, we present the operations scheduling problem as a combinatorial optimization problem. We present a simulated annealing based predictive scheduling algorithm for feed-forward control of the manufacturing system. For reactive operations scheduling, the closed loop configuration consists of the simulated annealing based predictive scheduler and the manufacturing system model in the feed forward path, and a reactive re-scheduler in the feedback path. Deviations of the observed sequenced operation timings from the scheduled timings, quantified by a cost functional, are used to trigger the re-scheduling loop. The capabilities of the scheme are demonstrated using a discrete event simulation of the Texas Instruments manufacturing unit.

1 INTRODUCTION

Operations scheduling in a flexible manufacturing system has been studied by many researchers in the past few years. Finding computationally feasible solutions for this NP-complete problem has been of continuous interest in the industrial engineering, operations research and manufacturing technology communities. Various approaches have been suggested notably, expert systems, neural network methods, heuristics based systems and other combinatorial optimization based methods. In general, these methods generate schedules assuming ideal operating conditions, i.e. no machine failures, constant setup and processing times etc., and hence can be categorized as predictive scheduling. The stochastics of a real world manufacturing system make the problem of scheduling even harder.

In this paper, we present a rule-based reactive scheduling scheme, which uses a predictive scheduler, in conjunction with a manufacturing system model, and a threshold triggered, feedback re-scheduler for on-line scheduling of the manufacturing system. Interested reader can refer to [1, 5] for complete description of the work presented here. The organization of the paper is as follows: In Section 2 we present description of the TI manufacturing unit, based on which we develop our general manufacturing system model in Section 3. Section

*This work was supported by NSF Engineering Research Centers Program NSFD CDR 88003012, and by a grant from Texas Instruments, Inc.

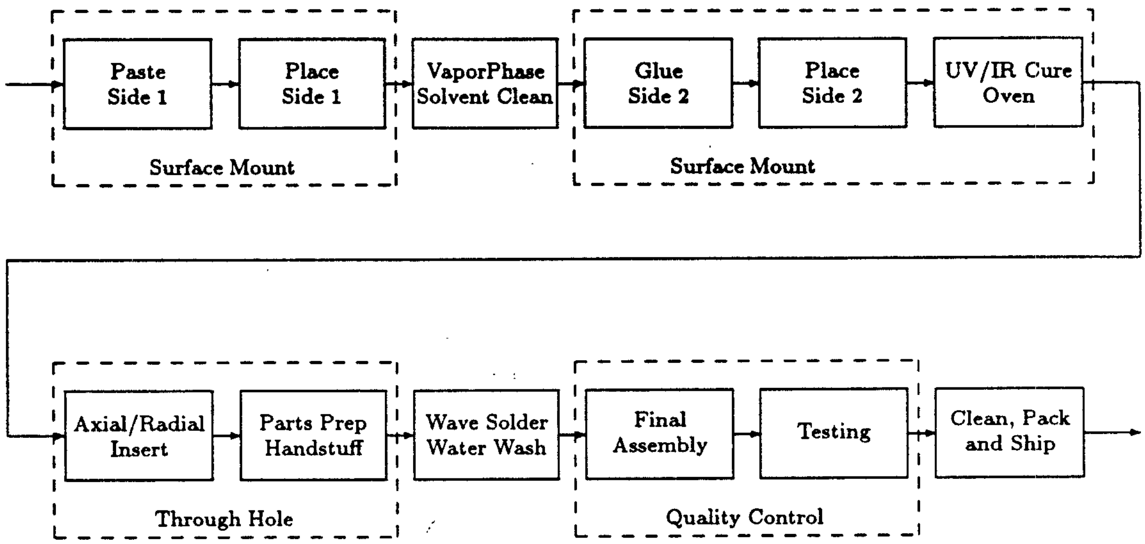

Figure 1: A typical process flow

4 describes the predictive scheduler, and Section 5 presents the algorithm used to trigger rescheduling. This will be followed by presentation of results on the example manufacturing system and concluding remarks in Sections 6 and 7, respectively.

2 TI Custom Manufacturing Unit

The Texas Instruments PC board assembly facility at Johnson City, TN assembles circuit boards for external customers and also for their Industrial Automation division. The Custom Manufacturing Business consists of three main units: surface mount, through-hole, and quality control. Both single and double sided boards are manufactured at this facility. A typical process flow for a double sided board is shown in Figure 1. For a single sided board the second surface mount operation is not needed. The surface mount unit is fully automated and has a lower operational cost than the through hole and quality control units which are semi-automated. The installed capacity of the surface mount unit is less than the other two units and is generally the bottleneck region in board process flow. Therefore, we shall focus on scheduling the surface mount unit.

The surface mount unit consists of six machine cells called SMT lines. Of these six lines, lines 1, 2, 3, and 4 are dedicated to top-side processing while the remaining two (5 and 6) are exclusively used for bottom side processing. The layout of the surface mount unit is shown in Figure 2. The top side processing lines, each consist of a screen-printer machine and a pair of component placement machines, while each of the remaining two lines have a glue-machine and a component placement machine followed by an Ultraviolet-Infrared Cure machine. The four SMT lines, 1 through 4, are connected through a conveyor system to the vapor phase and solvent clean machines. The boards, in general, are processed top side first on one of the four SMT lines and after being cleaned, are transported, via trolleys, to lines 5 or 6 for bottom side processing. The processing of the boards is continuous in each line, but the topside-processed and solvent cleaned boards are transported for bottom side processing in batches. Thus, the system has a mix of *continuous* and *batch flow* processing, which makes the scheduling job even tougher.

Each board has a *principal process flow* and a set of *alternate process flows*, which are determined by the characteristics of the parts being used in production. For example, if many fine pitch parts are being used on a particular board, then it can be assembled only on one of the four SMT lines for topside and similarly can be processed only on one of the two lines for bottom side processing. Hence, we only have a principal process flow and no alternate flows for this particular board. On the other extreme, there are certain boards which can be manufactured on any of the lines. Thus, a double sided board which can be run on any of the 4 lines for topside processing and either of the two lines (5 or 6) for bottom side processing has 8 possible process flow configurations. We assign any one of those process flows to be the principal one and the rest are alternate flows.

The Custom Manufacturing Business produces about 50 different boards. Depending on the customer demand and parts availability, it handles 30 different orders, or batches, per week on an average. The number of operations to be scheduled for each batch varies between 3 and 6 depending on the number of sides to be processed. Therefore, in principle we have about 100 operations to be scheduled each week.

The concept of "board families" plays an important role in schedule generation. If two batches of two different boards, belonging to the same board family, are scheduled successively on the same surface mount line, the setup time for the succeeding operation is reduced to a fraction of the original setup time. This occurs when the two boards have similar sets of components to be used on that particular line and so only a partial teardown/setup is required. These reduced setup times lead to increased machine utilization and higher productivity.

In certain batch orders, it is possible, that the raw material is not available at the time of the order. Hence, the operations can only be scheduled after the parts are made available. All these additional constraints, i.e., non-availability of the raw parts, machine availability over time, board family based reduced set-up times, machine assignment for each operation, the large number of operations to be scheduled in a given time frame etc., makes optimal scheduling a formidable task.

Earlier the surface mount machines were scheduled manually. Parts availability and the customer demand date were the main criteria used for scheduling. The different operations for various batches were then scheduled appropriately, on the 6 lines, so as to meet the due-date requirements. Approximate board processing times and machine setup times were used to determine the production run for each batch. This process of schedule generation was time consuming and tedious. Moreover, the quality of the schedule was heavily dependent

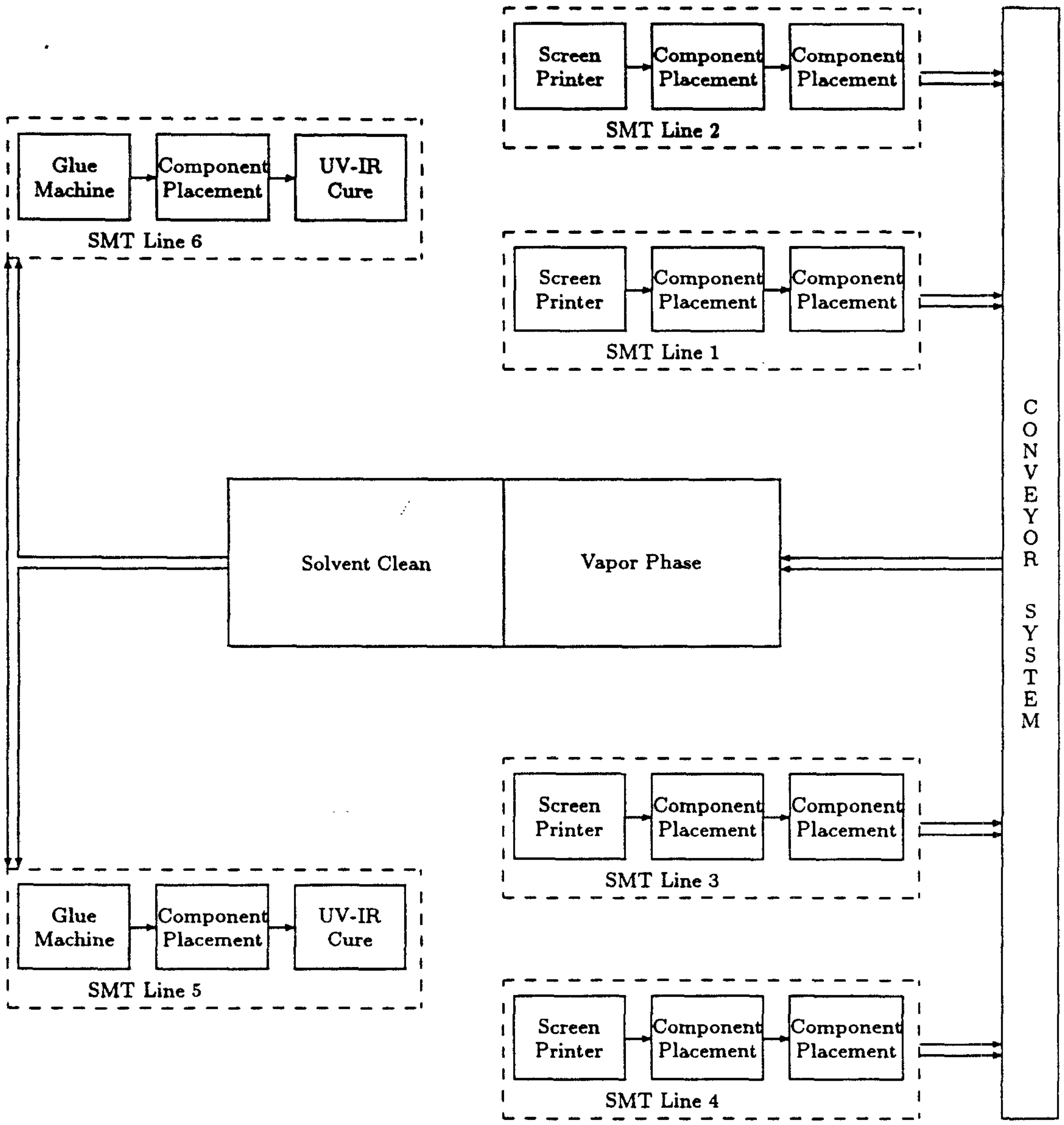

Figure 2: Surface Mount Unit

on the experience and expertise of the scheduler.

The Texas Instruments assembly unit provided us deep insight into the actual functioning of a real manufacturing system. The various operational issues, as discussed above, and their role in schedule generation provided us with a better understanding of the complexity and the intricate nature of flexible manufacturing system scheduling. The assembly unit provided a benchmark for the development of a general manufacturing system model. The modeling framework, along with the scheduling experience, were helpful in the development of a general purpose predictive scheduling software $\mathcal{ABES}$. The onsite visit, for the $\mathcal{ABES}$ installation phase, was further useful in understanding of the different control issues involved in manufacturing system and led to the development of this reactive operations scheduling algorithm.

3 Manufacturing System Model

Motivated by the flexible manufacturing system described above, we present a general, deterministic manufacturing system model. Though the model presented here is based on the TI assembly unit, it can be used to model different flexible manufacturing systems.

We model the factory as a set of physical machines $\mathcal{M} = \{m_1, m_2, ..., m_{|\mathcal{M}|}\}$. A machine $m_i \in \mathcal{M}$ can be a single physical machine or a virtual machine (ordered collection of machines which process parts in tandem). We will not distinguish between the concept of physical or virtual machine. For our purposes the machine is a processing station with a single inflow and outflow of parts. The manufacturing system produces a set of parts $\mathcal{P} = \{p_1,\ p_2,\ \ldots\ , p_{|\mathcal{P}|}\}$. To manufacture part p_i a sequence of operations, $< O^i_k >_{k=1}^{N_i}$, must be performed in order. That is, operation O^i_k must begin before operation O^i_{k+1}. As discussed in Section 2, it may be possible to perform a particular operation, say O^i_k, on any machine from a given subset of $\mathcal{M}$. On this basis, we can associate, to each part type p_i, a set of process flows $\mathcal{F}_i = \{f^i_1,\ f^i_2,\ \ldots\ f^i_{|\mathcal{F}_i|}\}$. A process flow f^i_l is defined as a sequence of machine operations $< {}_jO^i_k >_{k=1}^{|f^i_l|}$ where ${}_jO^i_k$ represents the k^{th} prototype operation in the process flow for part p_i, performed on machine m_j.

Let $\mathcal{O}$ denote the set of all operations, and let $\mathcal{SO}$ denote the set of finite sequences of distinct operations. The parts are produced in batches and we may have to schedule several batches for each part. Hence, there is also a set of *batches*, $\mathcal{B} = \{b_1, b_2, ..., b_{|\mathcal{B}|}\}$. Each batch contains many instances of a single part. For each batch b, of device i, there is a sequence of *operation instances*, denoted $< (b, O^i_k) >_{k=1}^{N_i}$; (b, O^i_k) represents operation O^i_k acting on all parts in batch b. We denote the set of operation instances as $\mathcal{OI}$, and the set of finite sequences of operation instances as $\mathcal{SOI}$. The selection of a particular process flow for a given batch is equivalent to the machine assignment for each constituent operation instance of that batch. Further, each batch b_i has an earliest start time, a due time, and a priority chosen from a finite set $\mathcal{W}$.

Each operation has a machine on which it must be performed and involves a *principal setup time*. The idea of *method family* is to group different part types into "families". Based on the operation sequencing on a machine, if two operations of the same *method family* are performed in succession on a machine, the actual time for a setup change is a fixed fraction of the principal setup time. This allows for the reduced setup representation discussed in section 2.

There are two ways to transfer parts from one operation to the next. In one case, the entire batch is processed by the first operation before it begins at the next. In the other, parts are "pipelined" from one operation to another. In this case, the second operation begins after a short delay, called the *transfer delay.* When batch processing is used instead of "pipelining," we say the transfer delay has the value BATCH.

In Appendix A, we present this formally in terms of logical relations, which are, for our purposes, functions mapping one set into another. We define several operators to make the notation concise.

4 Predictive Scheduler

Based on the manufacturing system model presented in the previous section, we formalize the concept of a schedule as a single sequence of operations. We present a general performance measure for this single sequence schedule representation, and then use it to formulate the job shop scheduling problem as an optimization problem.

4.1 Schedule definition

A *schedule* is an assignment of a process flow to each batch (the routing problem), and a sequence of operation instances to each physical machine (the sequencing problem). We identify the schedule, S, with a graph, $\mathbf{G}_S = (\mathbf{N}, \mathbf{E}_S)$, as follows. The vertices are all operation instances, $\mathbf{N} = \mathcal{OI}$. The set of edges, $\mathbf{E}_S$, comes from two sources. For each batch b, and each k, there is an edge in the graph from vertex $(b, O_k^{part(b)})$ to vertex $(b, O_{k+1}^{part(b)})$. For each machine, the sequence of operation instances assigned to it similarly defines a set of edges in the graph. A schedule is *feasible* if the graph so generated is acyclic. The precise definition of a schedule and its graph are as follows:
A *schedule*, denoted $S = (assign\_flow, mach\_op\_seq)$, is an ordered pair of relations which satisfies the following requirements:

$assign\_flow : \mathcal{B} \rightarrow \mathcal{F}$
  $assign\_flow(b) \in \mathcal{F}_{part(b)}$ for every batch $b \in S$

$mach\_op\_seq : \mathcal{M} \rightarrow \mathcal{SOI}$
  for any $(b, o) \in \mathcal{OI}$, there exists $m \in \mathcal{M}$ with (b, o) $\underline{in}$ $mach\_op\_seq(m)$

Notation:
  $\mathcal{S}$ = the set of feasible schedules.
  $\mathbf{N} = \mathcal{OI}$ = set of vertices
  $\mathbf{E}_S \subseteq \{(u, v)$, where $u, v \in \mathcal{OI}\}$ = directed edges for schedule S.

The graph $\mathbf{G}_S = (\mathbf{N}, \mathbf{E}_S)$ associated with schedule S satisfies:

(1) for $(b, o), (b, p) \in \mathcal{OI}, p$ $\underline{follows}$ o $\underline{in}$ $op\_seq(part(b)) \Longrightarrow ((b, o), (b, p)) \in \mathbf{E}_S$

(2) for $m \in \mathcal{PM}, (b^1, o^1), (b^2, o^2) \in \mathcal{OI}$,
(b^2, o^2) $\underline{follows}$ (b^1, o^1) $\underline{in}$ $mach\_op\_seq(m) \Rightarrow ((b^1, o^1), (b^2, o^2)) \in \mathbf{E}_S$
(3) Every edge in $\mathbf{E}_S$ is obtained from either (1) or (2)

As described earlier, a schedule consists of a method assignment and a sequence of operation instances for each physical machine. It is algorithmically and conceptually simpler to consider an alternative representation for the second part. Instead of using a separate sequence of operation instances for each machine, we use a single sequence containing all the operation instances. The space, $\mathcal{S}$, of feasible schedules is, then, replaced by $\mathcal{S}_{new} \equiv \mathcal{F} \times \mathcal{S}_n$, where $\mathcal{S}_n$ is the group of permutations of $n = |\mathcal{OI}|$ operation instances, and $\mathcal{F}$ is the union of sets of process flows. Our new representation for a schedule $S = (assign\_flow, mach\_op\_seq)$ is $S_{new} = (assign\_flow, \sigma)$, where $\sigma \in \mathcal{S}_n$ is a sequence of the operation instances.

We can determine σ from S, albeit nonuniquely, as follows. Form the graph $\mathbf{G}_S$ for S. Let α be an empty string of vertices of $\mathbf{G}_S$. Since S is feasible, $\mathbf{G}_S$ is acyclic, and there is a vertex which has no incoming arcs. Delete this vertex from the graph, and append it to the end of α. Also delete all of its outgoing arcs. The graph remaining is still acyclic. This process may be repeated until all vertices have been transferred to α. Now the sequence in α is a permutation of the elements of $\mathcal{OI}$, with the property that if there was a path from vertex A to vertex B in $\mathbf{G}_S$, then A appears earlier in α than does B.

The sequence of operation instances for a given batch, b, may be found from such a permutation by deleting from the list those operation instances belonging to a different batch. We call the sequence remaining the *filter* of α onto batch b, and denote this $filter_b(\alpha)$. Note that we must have:

$$filter_b(\alpha) = < (b, O_k^{part(b)}) >_{k=1}^{N_{part(b)}}$$

That is, the order of the operation instances must match that specified by $op\_seq(part(b))$. Similarly, the sequence of operation instances for physical machine m may be found by deleting any operation instance (b, o) not scheduled on machine m.

To illustrate this point consider a simple two machine (m_1 and m_2), two batch (b_1 and b_2) problem. Each batch consists of two operations. In batch b_1 the operations are sequenced first on m_1 and then on m_2; the reverse order is used by batch b_2. If operation ${}_bo_i^m$ represents the ith operation of batch b to be performed on machine m, then, for the above example, a possible schedule operation sequence is given by $< {}_1o_1^1, {}_2o_1^2, {}_1o_2^2, {}_2o_2^1 >$. Since operations ${}_1o_1^1$ and ${}_2o_2^1$ are for the same machine, this schedule operation sequence indicates that ${}_1o_1^1$ must complete before ${}_2o_2^1$ begins. This schedule operation sequence and the corresponding derived machine schedules are shown in Fig 3. The arrows denote the operation sequence for each batch. The same schedule often can be derived from other schedule operation sequences, such as $< {}_1o_1^1, {}_2o_1^2, {}_2o_2^1, {}_1o_2^2 >$ for our example.

The times when operation instances begin and end can be computed from the schedule. Each operation is assumed to begin as soon as all its preceding operations have taken place. The algorithm for computing these times, along with the associated sub-algorithms, are presented in Appendix B.

A performance measure for a schedule can then be computed from the start and finish times. Any performance measure which is computable from the start and finish times of all operations is permitted.

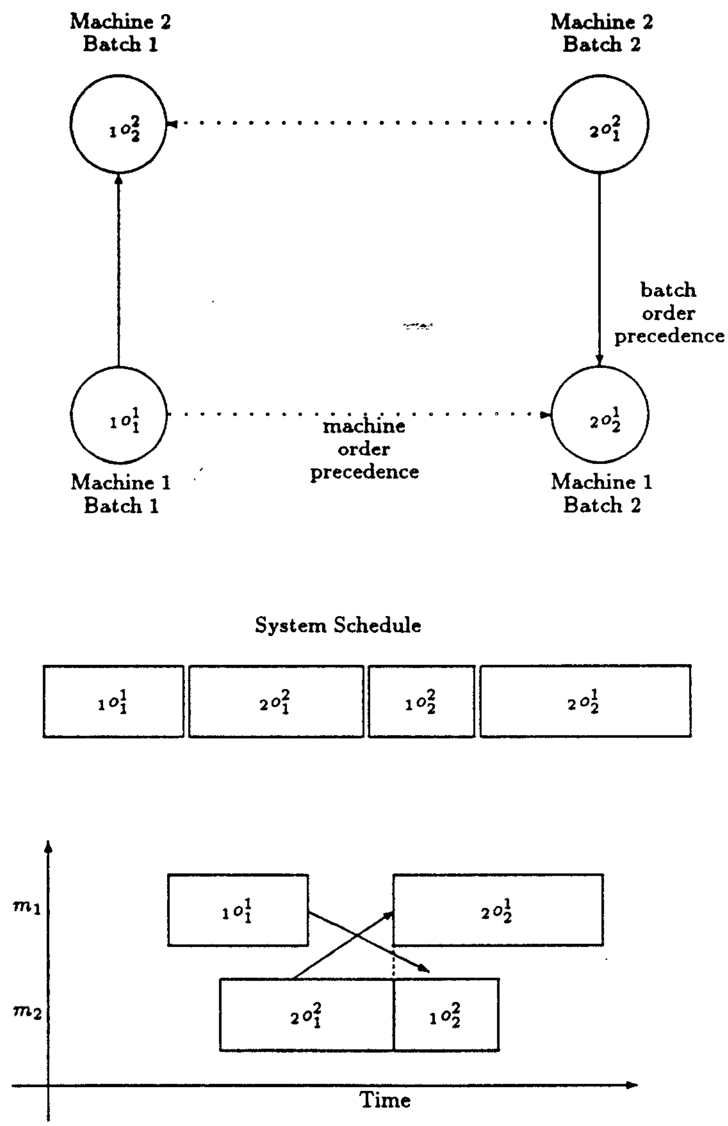

Figure 3: Derived machine schedules.

Our optimization problem is to **maximize the performance measure over the set of feasible schedules**, or equivalently, to **minimize the associated cost** i.e.:

$$\mathcal{C}^{\star} = \arg\min_{S \in \mathcal{S}} \mathcal{C}(t_s, t_f)$$

where

$\mathcal{TV} = \{$ all maps $\mathcal{OI} \rightarrow \Re\}$	$\mathcal{TV}$ stands for Time Vector.
$t_s : \mathcal{OI} \rightarrow \mathcal{TV}$	Operation start time.
$t_f : \mathcal{OI} \rightarrow \mathcal{TV}$	Operation finish time.
$\mathcal{C} : \mathcal{TV} \times \mathcal{TV} \rightarrow \Re$	This is the cost function.

4.2 Simulated Annealing Based Optimization

In the previous section, we represented a schedule as a single sequence of operations. The schedule optimization problem was posed as a maximization of a performance measure (minimization of an associated cost) over the set of feasible schedules. This optimization problem, and scheduling problems in general, are part of a larger group of *combinatorial optimization* problems. Combinatorial optimization problems deal with cost function maximization (or minimization) on a discrete problem domain.

Deterministic scheduling problems like many of the combinatorial optimization problems, are NP-complete. That is, the computational complexity (and hence the computation time) grows exponentially with the problem size. As a consequence, practical NP-complete problems, including our scheduling problem, can not be solved exactly in a reasonable time. Possible methods of finding an exact solution include exhaustive search, dynamic programming and branch-and-bound methods. These methods are based on enumeration of the possible solutions. In case of exhaustive search, as the name suggests, all solutions are evaluated to obtain a globally optimal solution. The other two cases involve a restricted search procedure. On the basis of the problem structure, certain solutions are eliminated from the search process. This leads to a reduction in the search domain size and reduction in computation time. These exact solution techniques are not very effective for solving practical problems, due to the large search domains associated to those problems.

Heuristic methods rely on finding sub-optimal solutions and then verifying the closeness (in terms of appropriate probabilistic bounds) of the solution to the "unknown" optimal solution. Rule-based expert systems, simulated annealing etc. are some of the popular heuristics based combinatorial optimization techniques.

Rule based expert systems use problem specific heuristics for reduction of the size of the problem domain. The efficiency of these techniques is heavily dependent on the selection of the heuristics or rules. Moreover, the problem based heuristic specification does not allow for a general solution method.

Simulated annealing uses stochastic search in a deterministic problem domain. The iterative search procedure is based on selection of trial solutions, and then accepting (or rejecting) the trial solutions based on a probabilistic structure. The probabilistic structure tends to move the search process towards a global optimum, without getting entrapped in a local optimum. Simulated annealing provides good sub-optimal solutions, but at the cost of long execution times.

Simulated annealing has been used extensively for solution of combinatorial optimization. It has been used for solving circuit layout and routing optimization problems. The extensive literature, general acceptance of the technique, and the natural adaptation for a two level process (discussed later in this section) makes it an ideal choice for our problem. Although simulated annealing methods have been applied to job shop scheduling problems [6], our approach is technically different and more general. We will not get into the detailed theoretical analysis of simulated annealing. The interested reader can refer to [5] and the references mentioned therein. We are more interested in using this technique for solving our optimization problem.

There are two major components to simulated annealing based optimal solution search. The first component is the *neighbourhood* definition. Simulated annealing relies on random selection of a trial solution from the set of neighbouring solutions. The second aspect is the choice of appropriate performance measure (or cost function) for the solution, which needs to be maximized (minimized). In the remainder of this section, we will present the performance measure, the neighbourhood structure used, and other implementational details.

In operations scheduling, performance measures of interest include due date performance, work in process inventories, and machine utilization. In our implementation, we permit a cost function including each of these elements, and we use a variation of "simulated annealing" to optimize it. The scheduling problem we consider has two related but distinguishable components, namely routing and sequencing. For this reason a two level stochastic descent method is natural.

Simulated annealing treats each schedule as a state in a discrete time finite state Markov chain. The transition probabilities of the Markov chain are selected so that the state drifts toward the lower cost schedules.

The simulated annealing algorithm uses a Markov process originally described by Metropolis [2]. An initial state (schedule), X_0, is chosen at random. Next, a nearby "trial" state, X_1^{trial}, is chosen by some probabilistic rule, and the two are compared. If the "trial" state has lower cost, it is accepted, with probability 1, as the new state of the Markov chain. Otherwise it is accepted with a lower probability determined by the relative costs. More precisely,

$$\begin{aligned}
\Pr\{X_{k+1} = X_{k+1}^{trial} | X_{k+1}^{trial}, X_k\} &= \begin{cases} 1 & \text{if } c(X_{k+1}^{trial}) < c(X_k) \\ e^{-[c(X_{k+1}^{trial}) - c(X_k)]/T_k} & \text{otherwise} \end{cases} \\
\Pr\{X_{k+1} = X_k | X_{k+1}^{trial}, X_k\} &= 1 - \Pr\{X_{k+1} = X_{k+1}^{trial} | X_{k+1}^{trial}, X_k\}
\end{aligned}$$

Here the value, T_k is called the "temperature," and the sequence $\{T_k\}_{k=0}^{\infty}$ is called the temperature schedule. If the trial state is only accepted when it has lower cost, the algorithm will remain in any local minimum it may find. This is the case when the temperature is zero. A nonzero temperature, however, gives the Markov chain the opportunity to escape from local minima. We will refer to a Markov process evolving in this manner as a Metropolis process. Simulated annealing is a special metropolis process in which $T_k \to 0$ slowly, so that the state of the Markov chain converges in probability to the collection of states with globally minimum cost[3]. In practice, since we can keep track of the best state visited so far, we are more interested in having ever reached a minimum than reaching it and staying there.

As mentioned earlier, we have a two level Metropolis process, corresponding to the two components of schedule generation. The Markov chain $< X_k >_{k=0}^{\infty}$ has $\mathcal{S}_{new}$ as its state space. We call $< X_k >_{k=0}^{\infty}$ the *high level* Metropolis process, and we define a so-called *low level* process to generate the random variable X_{k+1}^{trial}. We generate the initial state S_0^k, for the low level Metropolis process $< S_i^k >_{i=0}^{N}$, by using a different process flow assignment, but the same permutation of $\mathcal{OI}$ as X_k. The subsequent states of the Markov process, $< S_i^k >_{i=1}^{N}$ are generated using the same flow assignment but the permutation of $\mathcal{OI}$ changes.

A simple explanation on how the process evolves is as follows: Generating the trial schedule, S_{k+1}^{trial} simply involves perturbing S_k. As discussed in Section 3, each schedule can be represented by a global operation sequence. To perturb the schedule, we must permute the corresponding sequence. We do this by randomly selecting a short subsequence and a new position in the sequence which remains. Then the subsequence is reversed and moved to the new position.

The resulting global sequence may not satisfy all batch precedence constraints. To resolve this potential violation, each offending operation is swapped with the (unique) operation in the same batch which does not violate the batch precedence constraint. Finally, when all batch precedence constraints are satisfied, the start and finish times of all operations are computed using the rule that each operation begins as soon as its predecessors have completed (with an appropriate adjustment for setup time).

This heuristic method of generating trial schedules is derived, in an intuitive sense, from the actual practice of manual schedule generation. The idea of swapping operation, and then comparing the two resultant schedules, is consistent with the actual scheduling. There the scheduler is using his experience to evaluate and compare the two schedules, here the quantitative performance measure is being used for comparison. Hence, this adhoc procedure of *neighbourhood* definition provides good "trial" schedules.

The start and finish times are used to compute costs of tardiness, work-in-process, and finish goods storage. Based on the cost differential, a decision is made to accept or reject the trial state. Finally, after N such steps, we set $X_{k+1}^{trial} = S_N^k$. This method is shown to converge to the optimum in [4]. We stop the iterative procedure after a certain number of iterations or when cost stops improving, which results in a schedule which is good, though sub-optimal.

The initial schedule generation is done on the basis of addition of new batches to an empty schedule. The *operation instances* of the new batches are inserted in the schedule for minimum cost change. The complete algorithm is given in Appendix C.

5 Reactive Scheduler

The block diagram for reactive scheduling scheme is shown in Figure 4. The desired optimal schedule $\mathbf{S}$, generated by the scheduler, and the actual schedule from the model $\widehat{\mathbf{S}}$, are compared by the controller. The deviation between the two schedules is measured on

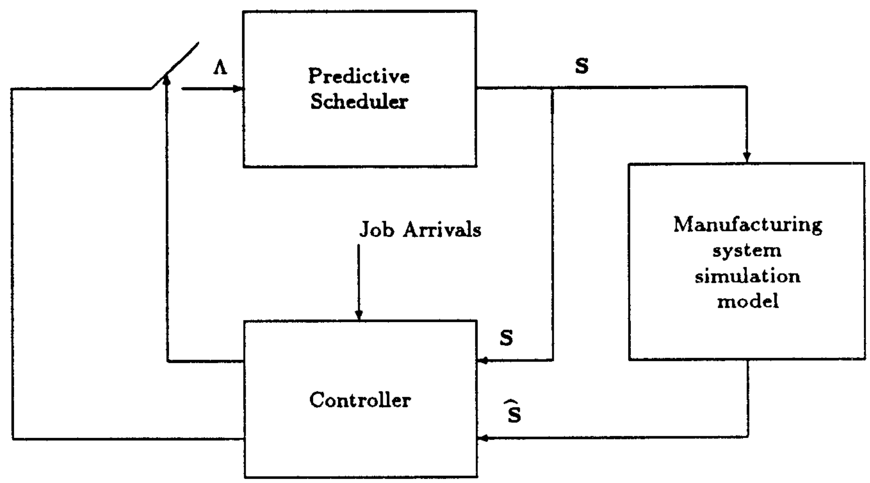

Figure 4: Block Diagram for Reactive Operations Scheduling

the basis of some cost criteria and a rule-based, threshold triggered scheme is used for rescheduling. At each rescheduling instant new jobs can be added to the schedule; this leads to on-line reactive scheduling of the manufacturing system. In Section 4, we presented details of the predictive scheduler. In this section, we will present details of the remaining two components in the closed loop and describe the closed loop control algorithm.

We have used the software package QNAP (Queuing Network Analysis Package) to develop a manufacturing system simulation model. Schedule $\mathbf{S}$ is input to the manufacturing system simulation model. (In the online system, the simulation model would be replaced by the actual factory.) The simulator has a central controller which keeps track of the batch operation precedence and also the machine ordering. The machines are modeled as single server queues and the batches are routed according to the pre-defined process flows. The simulator incorporates the various random manufacturing system features namely, short term/long term failures, random processing/setup times etc. The optimal schedules generated by the scheduler are used as an input to the simulator. The sequencing in the predictive schedule $\mathbf{S}$ is preserved in the simulation. However, the actual operation times, which differ from the scheduled time because of random processing and setup times, define the actual schedule $\widehat{\mathbf{S}}$.

In our simulation model, we allow operation times to be random (exponentially distributed). The machines are modeled as single server queues. We allow the machines to be either single physical machines or virtual machines (sequence of physical machines processing parts in tandem). Each machine has three different modes of failure. Failure Mode 1 (FM1) allows for random repair time and is more frequent than FM2 or FM3. FM2 and FM3 represent more drastic failures and have larger deterministic repair times. The difference between the three failure modes, is the severity of failure and the probability of occurrences. The severity of the failure can be translated into increase in repair times. For our simulation model, we let the repair times, for FM2 and FM3, to be 24 and 48 hrs respectively. The repair time for FM1 is a normal random variable, with mean 1 hr. The probability of failure for each mode is incremental with respect to number of operations processed since the previous failure. This is consistent with the machine failure probabilities in an actual manufacturing system.

As discussed in Section 4, each schedule has an *active time* associated to it. The rescheduling algorithm is based on a rule based, threshold triggered update of the active time and the manufacturing system status. In the remainder of this section, we will present the details of the different update rules and the rescheduling loop shown in Figure 4.

The predictive schedule $\mathbf{S}$ is used as an input to the manufacturing system simulation model. Since the batch order precedence is trivially maintained in the simulation, for notational convenience we omit the batch precedence ordering and use o_i^m to represent the i^{th} operation to be performed on machine m. In effect, we are re-labeling our operations on each machine and will carry out the analysis on the basis of individual machine schedules, i.e the machine order precedence, instead of the complete manufacturing system schedule. As defined earlier, $t_s(o_i^m)$ and $t_f(o_i^m)$ respectively, denote the start and the finish times for the operation o_i^m in the predictive schedule S. Now the schedule for machine m can be defined as:

$$\mathbf{S}_m \triangleq < o_k^m >_{k=1}^{N_m}; \quad t_s(o_k^m) \ < t_f(o_k^m) \leq \ t_s(o_{k+1}^m)$$

Then the overall schedule can be written as

$$\mathbf{S} = < \ \mathbf{S}_m >_{m=1}^{M}$$

In the following we use t to represent "real time", that is, the position in the actual schedule, $\widehat{\mathbf{S}}$. For each machine, there is a point in the predictive schedule corresponding to t in $\widehat{\mathbf{S}}$. This point is the "active time" corresponding to real time t, which we write $\tau_m(t)$. Precisely, suppose operation o^m is in process at time t in the actual schedule, $\widehat{\mathbf{S}}$, and suppose it is expected to complete at time $t + \delta$. Then the active time for machine m is given by

$$\tau_m(t) = t_f(o^m) - \delta.$$

Active times are used by the controller as indicated below.

Initially the active times on all machines are equal to the real time, say $\tau_m(t_0) = t = t_0$. Because of the various random features of the simulation, the two schedules differ in terms of the operation timings. To correlate the two schedules, at real time t ($t_0 \leq t$), we can

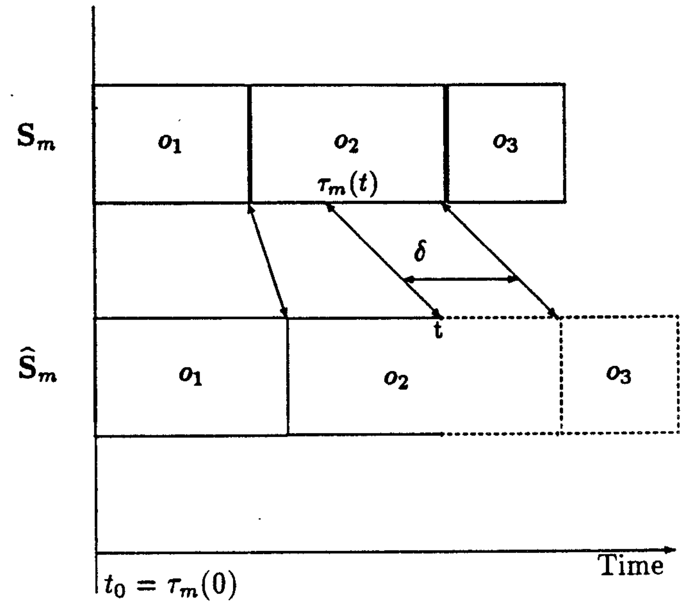

Figure 5: Active Times for a single machine

define the active time vector $\Lambda(t) = [\tau_1(t), \tau_2(t) \ldots \tau_M(t)]^T$ $(\Lambda(0) = [t_0, t_0 \ldots t_0])$. $\Lambda(t)$ represents the present status of $\widehat{\mathbf{S}}$ in $\mathbf{S}$. For illustration, Fig 5 shows the two time scales for a single machine schedule.

For each operation o^i_m, we associate a cost $c^i_m = c(t_s(o^i_m), t_f(o^i_m), \hat{t_s}(o^i_m), \hat{t_f}(o^i_m))$. This cost represents the penalty involved for the actual operation times to differ from the scheduled times. Based on the individual operation costs, the net schedule cost can be calculated as

$$C_r(\mathbf{S}, \widehat{\mathbf{S}}, t_0, t) = \sum_{m=1}^{M} \sum_{\substack{i=1 \\ t_0 \le \hat{t_f}(o^i_m) \le t}}^{N_m} c^i_m$$

The controller keeps track of the two schedules $\mathbf{S}$ and $\widehat{\mathbf{S}}$ and at the end of the each operation updates the net schedule cost $C_r(\mathbf{S}, \widehat{\mathbf{S}}, t_0, t)$. The rescheduling loop is triggered under the following conditions:

- $C_r(\mathbf{S}, \widehat{\mathbf{S}}, t_0, t) > C_T$. Here C_T is some predefined cost threshold.
- Failure Mode 2 on any machine
- Failure Mode 3 on any machine
- New Job arrivals

At each such rescheduling instant t, the controller polls in the current status $(\Lambda(t))$ of the manufacturing system schedule. This status is then input to the scheduler. If the rescheduling is necessary due to failure 2 or 3 on any machine, an extra input in the form of the operation o^m_f along with its associated start and finish times is sent to the scheduler.

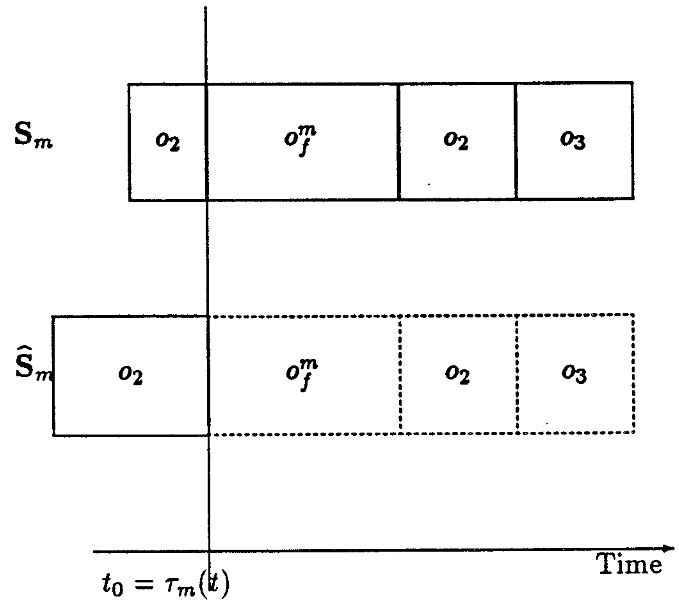

Figure 6: Schedule representation for machine failure induced rescheduling

The Scheduler uses the above information to update the present schedule $\mathbf{S}$. Past operations are discarded, and the current real time t becomes the new initial time $t_0 \leftarrow t$. Hence, $\tau_m(t) \leftarrow t$ also. The scheduler then allows the user to make changes (add/remove batches or operations) in the schedule $\mathbf{S}^1$. Next the simulated annealing based optimization is activated to optimize the remainder of the schedule. Since the manufacturing model is continuously simulating the schedule $\mathbf{S}$, the optimization is not done for the immediately scheduled operations. Basically we do not reschedule the operations scheduled to start between t_0 and $t_0 + T$, for some disable period $T > 0$. This allows for on-line optimization of the current schedule while the simulation is still running. (We do not literally run the simulation and the controller concurrently; this is a consideration for online implementation.) Once an optimal schedule is obtained it is input to the simulation model. In case of failure mode rescheduling, the failure operation o_f^m is treated as an operation with fixed processing time. Since in this case $t_s(o_f^m) = t_0$ the operation is not rescheduled and it is simulated as a repair time. Taking the schedule shown in Fig 5, if a failure FM2 or FM3 occurs at time t on machine m, then the two schedules $\mathbf{S}$ and $\hat{\mathbf{S}}$ after rescheduling are depicted in Fig 6. This closed loop process is carried out continuously and leads to reactive operations scheduling for manufacturing system. The complete process can be represented in a flow chart as shown in Fig 7.

The complete algorithm for the reactive scheduling based process control is as follows: At each rescheduling instant t, the time vector $\Lambda(t)$, which represents the manufacturing system status, is input to the controller. The additional information (if it is a failure

[1]To maintain compatibility with the notation, the operations are re-numbered from $1 \ldots N_m$ for each machine m

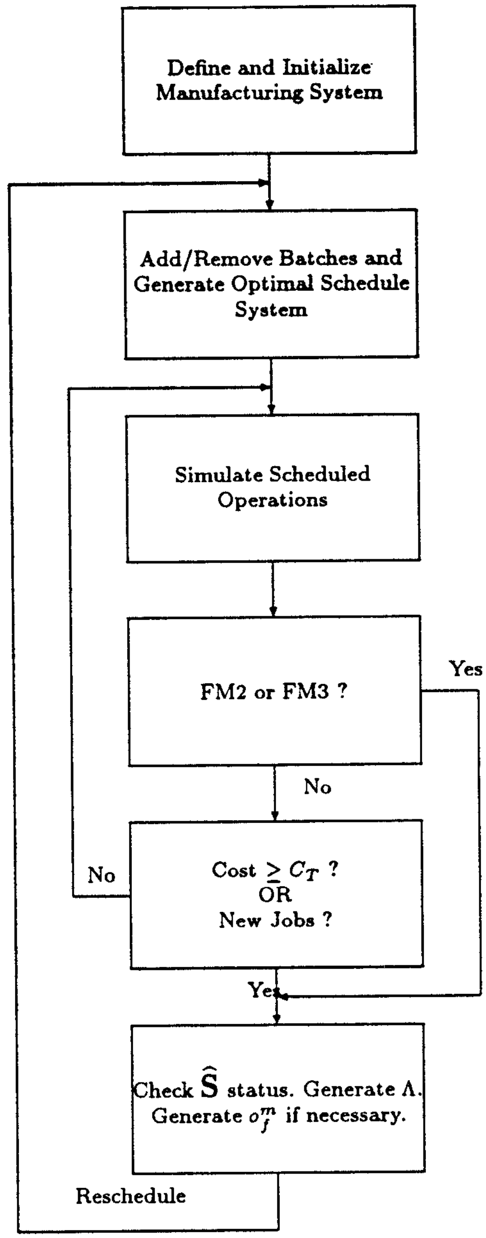

Figure 7: Rescheduling Algorithm Flow chart

triggered rescheduling) namely, dummy operations with their associated start and finish times, is also input to the scheduler. This manufacturing system status is then used to increment the active time, and hence the predictive schedule, by the scheduler. Using this updated schedule as the initial state, X_0, for the *high level* Metropolis process, the simulated annealing process is activated. The initial state S_0^k for the *low level* process, $< S_i^k >_{i=0}^{N}$, is generated using the following algorithms:

Initial state generation

algorithm $init\_state(S_{old})$
$\star$ $b = Random\_Select(S_{old})$
if $((\mathcal{F}_{part(b)} \setminus assign\_flow_{old}(b)) \neq \emptyset)$
$assign\_flow_{new}(b) = f_i^{part(b)} \in (\mathcal{F}_{part(b)} \setminus assign\_flow_{old}(b))$
else go to $\star$
endif
return(S_{new})
end_algorithm

Random batch select

algorithm $Random\_Select(S)$
return$(b_{random(1,|\mathcal{B}|)})$
end_algorithm

Thus, using the initial schedule S_0^k, the *low level* Metropolis process is generated, for N steps, using the procedure discussed in Section 4. The trial schedule X_{k+1}, for the high level process is set equal to S_N^k. The high level temperature schedule is used to determine acceptance/rejection of the trial schedule. This process is continued for the two levels and since the scheduler keeps track of the best schedule generated in any time interval, a good sub-optimal reconfigured schedule is generated in finite number of iterations. This process is repeated at each re-scheduling instant and leads to reactive operations scheduling based on-line control.

6 Results

In this section, we present a hybrid manufacturing system model, i.e., a model based on both batch[2] and continuous[3] mode processing. The manufacturing system model is again based on the Custom Manufacturing Unit at Texas Instruments, Inc., Johnson City, but has been suitably modified to incorporate the complete process flow. This test model will be used to present results on actual implementation of the reconfigurable control algorithms.

[2]processing a batch of a single part type and then switch to processing a batch of another part type.
[3]process different part types in the order they arrive

Figure 8 depicts the complete hybrid manufacturing system model. The model consists of 7 machines and is configured to process 4 different part types. The 7 machines with their corresponding descriptive, functional titles are as follows:

M/C-1: IC Placement.

M/C-2: Axial Insert.

M/C-3: Axial Sequence Insert.

M/C-4: Radial Insert.

M/C-5: Manual Test Station.

M/C-6: Burn-In Station.

M/C-7: Final Test Station.

M/C-1 is models the surface mount machines discussed in Section 2, while the next three machines, namely, Axial Insert, Axial Sequence Insert, and Radial Insert, are part of the process flow as shown in Figure 1. The last three stations, namely Manual test, Burn-in and Final test, are part of the quality control unit. These stations have parallel work-stations where many parts can be processed at any time instant. The test sequences for different part types are very similar and hence these machines (in the case of Manual Test station, human operators) can be assumed to be operational in a continuous processing mode.

For our analysis we will assume that the insert machines, namely, machines M/C-1 through M/C-4, operate in a batch mode and the various operations are scheduled accordingly on these machines. On completion of the last insert type operation, the batches are just queued into the buffer of the first test station, namely M/C-5. The various processing times and set-up times, at each of the machines, are adapted from actual data from the manufacturing unit. As the test stations are operational in a continuous mode and are the last operation in all process flow they need not be scheduled explicitly. Hence, we only consider the schedules for M/C-1 through M/C-4.

Machines M/C-2 and M/C-3 are both Axial Insert machines, only difference being the mode of insertion of components. In the Axial Sequence Insert (M/C-3) the parts are pre-sequenced on a single part magazine and parts are sequentially accessed and inserted. In the case of M/C-2 the individual parts are available through a battery of part magazines and the part holder accesses each of the magazines for the respective parts. Hence, in principal these machines can be used interchangeably with minor reconfiguration. This allows for the specification of alternate flows for each of the part types which are processed through either of the machines. The alternate flows for each part types are depicted by dashed lines in Figure 8.

We used the predictive scheduler to generate a working schedule, for a given set of job orders (Figure 9). Figures 10 and 11 represent the initial schedule, generated by the *insert algorithm* discussed in Section 4. The next step was to obtain an optimal schedule using the simulated annealing based optimization process. Figures 12 and 13 show the optimal schedule. On comparison of the two schedules, we note that the two particular batches of

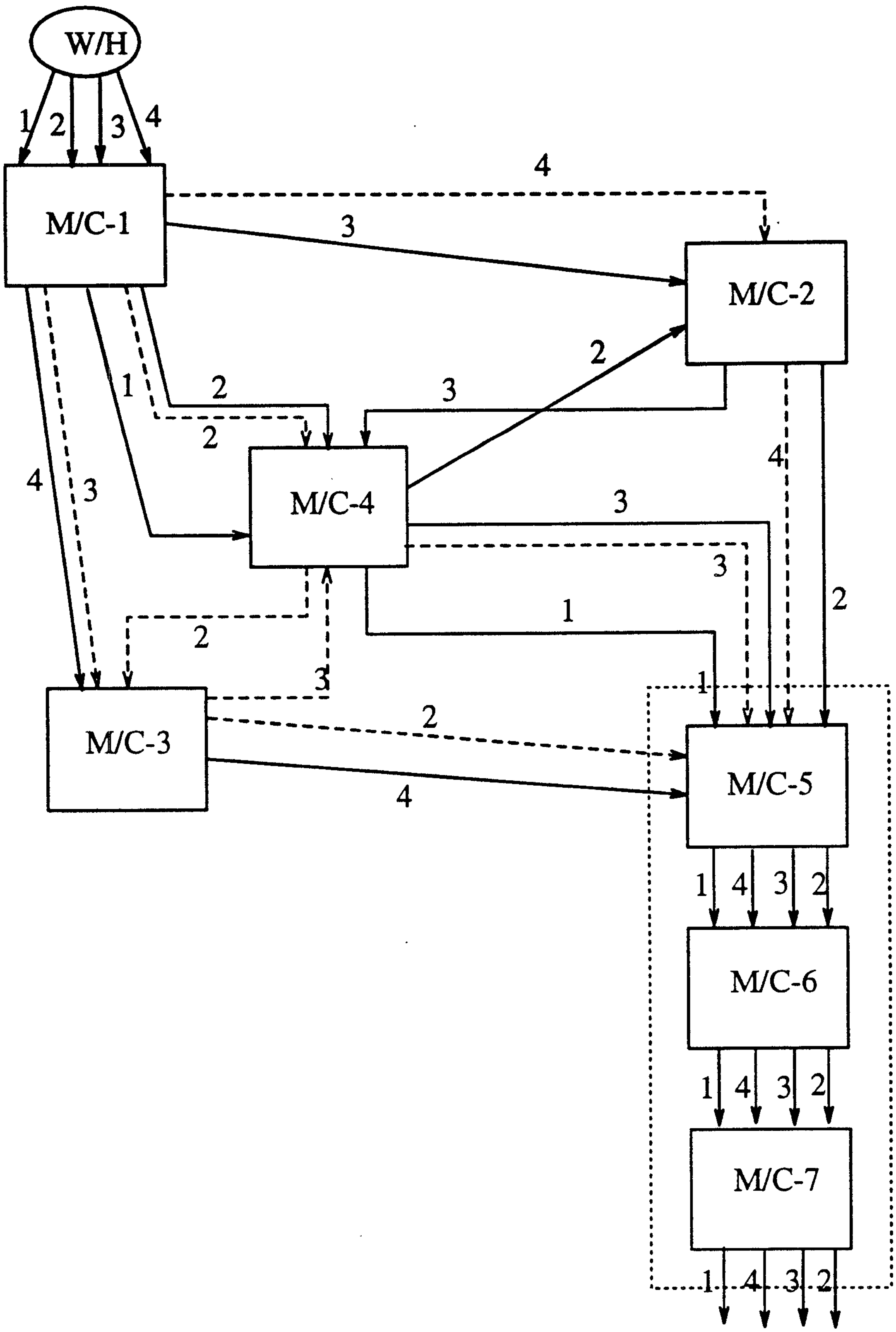

Figure 8: Hybrid Manufacturing System

PCB 3, namely BATCH ID 011 and BATCH ID 004, and also BATCH ID 001 of part type PCB 4, have been scheduled using their alternate process flows in the optimized schedule. This alternate flow assignment was possible due to the ***higher level*** Metropolis process. Figure 16 shows the trace of the cost and the temperature versus number of iterations for the annealing based optimization process. The sharp increase/decrease in the temperature graph represents the artificially forced, manual increase/decrease in the temperature. The increase in the temperature allows the optimization process to accept higher cost schedules with a larger probability and hence allows the process to "jump" out of the neighbourhood of a local optimum and hence converge to a global optimum. The corresponding cost trace during that period is, as expected, more "noisy". The schedule statistics for the initial and the optimized schedule are shown in Figures 14 and 15 respectively. The optimization process has reduced the number of tardy batches from 4 to 3 and also the average tardiness from 27 hrs/batch to 10.3 hrs/batch.

Once the initial optimal schedule was generated, the schedule was simulated using the QNAP2 based simulation model. The trace of the simulation for the first run (and also the subsequent second run after rescheduling) is shown in Figure 17. The simulation is stopped on failure type 2 on machine M/C-1. The simulation program then outputs a status report to the controller. The controller then updates the active time and also the status of different operations. The operation timings for the completed operations, for the first run are shown in Figure 19. The controller activates the optimization process again for an optimal schedule generation. The new schedule (shown in Figures 21 and 22) is now again input to the simulation program. The simulation trace and the completed operation timings for the second run are shown in Figures 17 and 20 respectively. The rescheduling cost (C_r) trace for the manufacturing system is shown in Figure 18. In both simulation runs, the rescheduling loop was triggered due to machine failures, but the machine operations were not rescheduled, using the sub-algorithm, as no alternate machines were available. At each such rescheduling trigger, new batches/orders could be inserted into the updated schedule and the process continued indefinitely. This leads to reactive operations scheduling of the manufacturing system.

7 Conclusions

In this paper, we presented a deterministic manufacturing system model. The operation based manufacturing system model was used to represent the system schedule as a single sequence of operations. This representation was useful in the development of the simulated annealing based optimization algorithm and also the subsequent implementation as a software package. The development of this software was helpful in better understanding of the manufacturing system modeling issues. This also provided insight into the control problems/issues in manufacturing systems. This led to the development of reactive operations scheduling algorithm.

Though the results presented in this paper, were based on the TI assembly unit, the methodology (and the software) can be easily used for other manufacturing systems. The combination of heuristics and analytical techniques provided a reasonable compromise between optimal, intractable solution methods and sub-optimal, expert systems approach for

Batch ID	Part Type	Batch Qty	Due Date
001	PCB 4	425	10/16/92
002	PCB 3	350	10/16/92
003	PCB 2	200	10/12/92
004	PCB 3	450	10/16/92
005	PCB 4	350	10/14/92
006	PCB 2	500	10/13/92
007	PCB 1	250	10/14/92
008	PCB 4	500	10/16/92
009	PCB 1	500	10/14/92
010	PCB 2	200	10/16/92
011	PCB 3	400	10/16/92
012	PCB 1	750	10/15/92

Figure 9: Job Orders for the Manufacturing System

solving this difficult problem. It is difficult to evaluate our schedules in "monetary" terms; the real evaluation is possible only by the actual user. The increase in throughput, reduced inventory levels, and reduced tardiness, quantified by the appropriate cost functional, validates the usefulness of our scheme.

Appendix A:

Operators

$\underline{in} : \mathcal{O} \times \mathcal{SO} \rightarrow \{$ true, false $\}$

$(o \ \underline{in} \ seq) =$ true $\Leftrightarrow seq = < o_1, o_2, ..., o_n > \bigwedge o = o_i$ for some i

$(\underline{follows}, \underline{in}) : \mathcal{O} \times \mathcal{O} \times \mathcal{SO} \rightarrow \{$ true, false $\}$

$(o_i \ \underline{follows} \ o_j \ \underline{in} \ seq) =$ true $\Leftrightarrow seq = < k_1, k_2, ..., k_n > \bigwedge k_p = o_j \bigwedge k_{p+1} = o_i$ for some p.

Again, a similar definition applies for

$(\underline{follows}, \underline{in}) : \mathcal{OI} \times \mathcal{OI} \times \mathcal{SOI} \rightarrow \{$ true, false $\}$

Distinction between the operators with identical names will be clear from context.

Relations

$op\_seq \ \mathcal{D} \rightarrow \mathcal{SO}$	The sequence of operations used to manufacture a part, subject to $o \ \underline{in} \ op\_seq(b) \bigwedge o \ \underline{in} \ op\_seq(a) \Rightarrow a = b$
$part : \mathcal{B} \rightarrow \mathcal{D}$	The part type of a given batch.

PCB 01

Operation ID	Machine ID	Setup Time	Batch Qty	Start Time	Finish Time
Batch ID: 007					
0099	(parts)	0	250		18:38:08 (10/12)
0100	M/C-1	30	250	22:39:48 (10/12)	00:09:05 (10/13)
0101	M/C-4	30	250	17:10:45 (10/13)	20:09:19 (10/13)
				DUE:	23:00:00 (10/14)
Batch ID: 012					
0074	(parts)	0	750		18:38:08 (10/12)
0075	M/C-1	30	750	05:32:11 (10/14)	10:00:02 (10/14)
0076	M/C-4	30	750	20:02:11 (10/15)	04:57:54 (10/16)
				DUE:	23:00:00 (10/15)
Batch ID: 009					
0105	(parts)	0	500		18:38:08 (10/12)
0106	M/C-1	30	500	09:06:14 (10/13)	12:04:48 (10/13)
0107	M/C-4	30	500	20:09:19 (10/13)	02:06:28 (10/14)
				DUE:	23:00:00 (10/16)

PCB 02

Operation ID	Machine ID	Setup Time	Batch Qty	Start Time	Finish Time
Batch ID: 003					
0084	(parts)	0	200		18:38:08 (10/12)
0085	M/C-1	45	200	19:23:08 (10/12)	20:10:45 (10/12)
0086	M/C-4	30	200	13:06:28 (10/13)	16:40:45 (10/13)
0087	M/C-2	60	200	00:05:02 (10/14)	04:50:45 (10/14)
				DUE:	23:00:00 (10/12)
Batch ID: 006					
0095	(parts)	0	500		18:38:08 (10/12)
0096	M/C-1	45	500	20:10:45 (10/12)	22:09:48 (10/12)
0097	M/C-4	30	500	04:10:45 (10/13)	13:06:28 (10/13)
0098	M/C-2	60	500	12:10:45 (10/13)	00:05:02 (10/14)
				DUE:	23:00:00 (10/14)
Batch ID: 010					
0108	(parts)	0	200		18:38:08 (10/12)
0109	M/C-1	45	200	12:49:48 (10/13)	13:37:25 (10/13)
0110	M/C-4	30	200	02:36:28 (10/14)	06:10:45 (10/14)
0111	M/C-2	60	200	12:53:36 (10/15)	17:39:19 (10/15)
				DUE:	23:00:00 (10/16)

Figure 10: Initial Schedule (Part I)

```
******************************************************************************
************************** PCB 03 *******************************************
******************************************************************************
Operation Machine  Setup  Batch  Start                Finish
   ID        ID    Time    Qty   Time                 Time
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Batch ID: 002
 0080     (parts)    0    350                         18:38:08 (10/12)
 0081   M/C-1      120    350    19:33:36 (10/14)     22:20:16 (10/14)
 0082   M/C-2       20    350    03:33:36 (10/15)     11:53:36 (10/15)
 0083   M/C-4       45    350    05:42:54 (10/16)     18:12:54 (10/16)
                                             DUE:     23:00:00 (10/13)
Batch ID: 011
 0112     (parts)    0    400                         18:38:08 (10/12)
 0113   M/C-1      120    400    15:37:25 (10/13)     18:47:54 (10/13)
 0114   M/C-2       20    400    05:10:45 (10/14)     14:42:11 (10/14)
 0115   M/C-4       45    400    13:10:45 (10/14)     03:27:54 (10/15)
                                             DUE:     23:00:00 (10/14)
Batch ID: 004
 0088     (parts)    0    450                         18:38:08 (10/12)
 0089   M/C-1      120    450    18:47:54 (10/13)     22:22:11 (10/13)
 0090   M/C-2       20    450    14:42:11 (10/14)     01:25:02 (10/15)
 0091   M/C-4       45    450    03:27:54 (10/15)     19:32:11 (10/15)
                                             DUE:     23:00:00 (10/15)


******************************************************************************
************************** PCB 04 *******************************************
******************************************************************************
Operation Machine  Setup  Batch  Start                Finish
   ID        ID    Time    Qty   Time                 Time
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Batch ID: 005
 0092     (parts)    0    350                         18:38:08 (10/12)
 0093   M/C-1      150    350    00:52:11 (10/14)     05:02:11 (10/14)
 0094   M/C-3      120    350    08:52:11 (10/14)     21:22:11 (10/14)
                                             DUE:     23:00:00 (10/14)
Batch ID: 008
 0102     (parts)    0    500                         18:38:08 (10/12)
 0103   M/C-1      150    500    02:39:05 (10/13)     08:36:14 (10/13)
 0104   M/C-3      120    500    10:39:05 (10/13)     04:30:31 (10/14)
                                             DUE:     23:00:00 (10/15)
Batch ID: 001
 0077     (parts)    0    425                         18:38:08 (10/12)
 0078   M/C-1      150    425    12:30:02 (10/14)     17:33:36 (10/14)
 0079   M/C-3      120    425    21:22:11 (10/14)     12:32:54 (10/15)
                                             DUE:     23:00:00 (10/16)
******************************************************************************
******************************************************************************

Active Time of schedule: 18:38:08, Oct 12, 1992
```

Figure 11: Initial Schedule (Part II)

```
*****************************************************************************
************************** PCB 01 *******************************************
*****************************************************************************
Operation Machine  Setup  Batch  Start                 Finish
   ID        ID    Time    Qty   Time                  Time
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
Batch ID: 007
 0099      (parts)   0     250                         18:38:08 (10/12)
 0100   M/C-1       30     250   17:44:48 (10/13)      19:14:05 (10/13)
 0101   M/C-4       30     250   19:55:17 (10/14)      22:53:51 (10/14)
                                             DUE:      23:00:00 (10/14)
Batch ID: 000
 0074      (parts)   0     750                         18:38:08 (10/12)
 0075   M/C-1       30     750   08:21:14 (10/14)      12:49:05 (10/14)
 0076   M/C-4       30     750   22:53:51 (10/14)      07:49:34 (10/15)
                                             DUE:      23:00:00 (10/15)
Batch ID: 009
 0105      (parts)   0     500                         18:38:08 (10/12)
 0106   M/C-1       30     500   14:51:42 (10/14)      17:50:17 (10/14)
 0107   M/C-4       30     500   05:13:08 (10/16)      11:10:17 (10/16)
                                             DUE:      23:00:00 (10/16)


*****************************************************************************
************************** PCB 02 *******************************************
*****************************************************************************
Operation Machine  Setup  Batch  Start                 Finish
   ID        ID    Time    Qty   Time                  Time
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
Batch ID: 003
 0084      (parts)   0     200                         18:38:08 (10/12)
 0085   M/C-1       45     200   19:23:08 (10/12)      20:10:45 (10/12)
 0086   M/C-4       30     200   03:23:08 (10/13)      06:57:25 (10/13)
 0087   M/C-2       60     200   17:29:48 (10/13)      22:15:31 (10/13)
                                             DUE:      23:00:00 (10/12)
Batch ID: 006
 0095      (parts)   0     500                         18:38:08 (10/12)
 0096   M/C-1       45     500   20:10:45 (10/12)      22:09:48 (10/12)
 0097   M/C-4       30     500   06:57:25 (10/13)      15:53:08 (10/13)
 0098   M/C-2       60     500   22:15:31 (10/13)      10:09:48 (10/14)
                                             DUE:      23:00:00 (10/14)
Batch ID: 010
 0108      (parts)   0     200                         18:38:08 (10/12)
 0109   M/C-1       45     200   13:34:05 (10/14)      14:21:42 (10/14)
 0110   M/C-4       30     200   01:08:51 (10/16)      04:43:08 (10/16)
 0111   M/C-2       60     200   09:08:51 (10/16)      13:54:34 (10/16)
                                             DUE:      23:00:00 (10/16)
```

Figure 12: Optimized Schedule after 5000 iterations (Part I)

```
******************************************************************************
************************** PCB 03 *******************************************
******************************************************************************
Operation Machine Setup Batch Start               Finish
  ID        ID    Time   Qty  Time                Time
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Batch ID: 002
 0080      (parts)   0   350                      18:38:08 (10/12)
 0081   M/C-1      120   350  00:09:48 (10/13)    02:56:28 (10/13)
 0082   M/C-2       20   350  08:09:48 (10/13)    16:29:48 (10/13)
 0083   M/C-4       45   350  16:38:08 (10/13)    05:08:08 (10/14)
                                          DUE:    23:00:00 (10/13)
Batch ID: 011
 0112      (parts)   0   400                      18:38:08 (10/12)
 0113   M/C-1      120   400  02:56:28 (10/13)    06:06:57 (10/13)
 0114   M/C-3       20   400  10:56:28 (10/13)    20:27:54 (10/13)
 0115   M/C-4       45   400  05:08:08 (10/14)    19:25:17 (10/14)
                                          DUE:    23:00:00 (10/14)
Batch ID: 004
 0088      (parts)   0   450                      18:38:08 (10/12)
 0089   M/C-1      120   450  06:06:57 (10/13)    09:41:14 (10/13)
 0090   M/C-3       20   450  20:27:54 (10/13)    07:10:45 (10/14)
 0091   M/C-4       45   450  08:34:34 (10/15)    00:38:51 (10/16)
                                          DUE:    23:00:00 (10/15)


******************************************************************************
************************** PCB 04 *******************************************
******************************************************************************
Operation Machine Setup Batch Start               Finish
  ID        ID    Time   Qty  Time                Time
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Batch ID: 005
 0092      (parts)   0   350                      18:38:08 (10/12)
 0093   M/C-1      150   350  21:44:05 (10/13)    01:54:05 (10/14)
 0094   M/C-3      120   350  09:10:45 (10/14)    21:40:45 (10/14)
                                          DUE:    23:00:00 (10/14)
Batch ID: 008
 0102      (parts)   0   500                      18:38:08 (10/12)
 0103   M/C-1      150   500  01:54:05 (10/14)    07:51:14 (10/14)
 0104   M/C-3      120   500  21:40:45 (10/14)    15:32:11 (10/15)
                                          DUE:    23:00:00 (10/15)
Batch ID: 001
 0077      (parts)   0   425                      18:38:08 (10/12)
 0078   M/C-1      150   425  12:11:14 (10/13)    17:14:48 (10/13)
 0079   M/C-2      120   425  12:09:48 (10/14)    03:20:31 (10/15)
                                          DUE:    23:00:00 (10/16)
******************************************************************************
******************************************************************************

Active Time of schedule: 18:38:08, Oct 12, 1992
```

Figure 13: Optimized Schedule after 5000 iterations (Part II)

```
********************************************************
*             SCHEDULE STATISTICS DISPLAY              *
********************************************************
Schedule Active Time: 18:38:08, Oct 12, 1992

Number of batches...
           in schedule :     12
           tardy       :      4
Mean Tardiness          :   9.0 Hours

Average tardiness of tardy batches :     26.9 Hours

COST    WIP ==>             0.0
  Tardiness ==>           537.5
  Inventory ==>             0.0


********************************************************
```

Figure 14: Initial Schedule Statistics

```
********************************************************
*             SCHEDULE STATISTICS DISPLAY              *
********************************************************
Schedule Active Time: 18:38:08, Oct 12, 1992

Number of batches...
           in schedule :     12
           tardy       :      3
Mean Tardiness          :   2.6 Hours

Average tardiness of tardy batches :     10.3 Hours

COST    WIP ==>             0.0
  Tardiness ==>           155.2
  Inventory ==>             0.0


********************************************************
```

Figure 15: Optimal Schedule Statistics

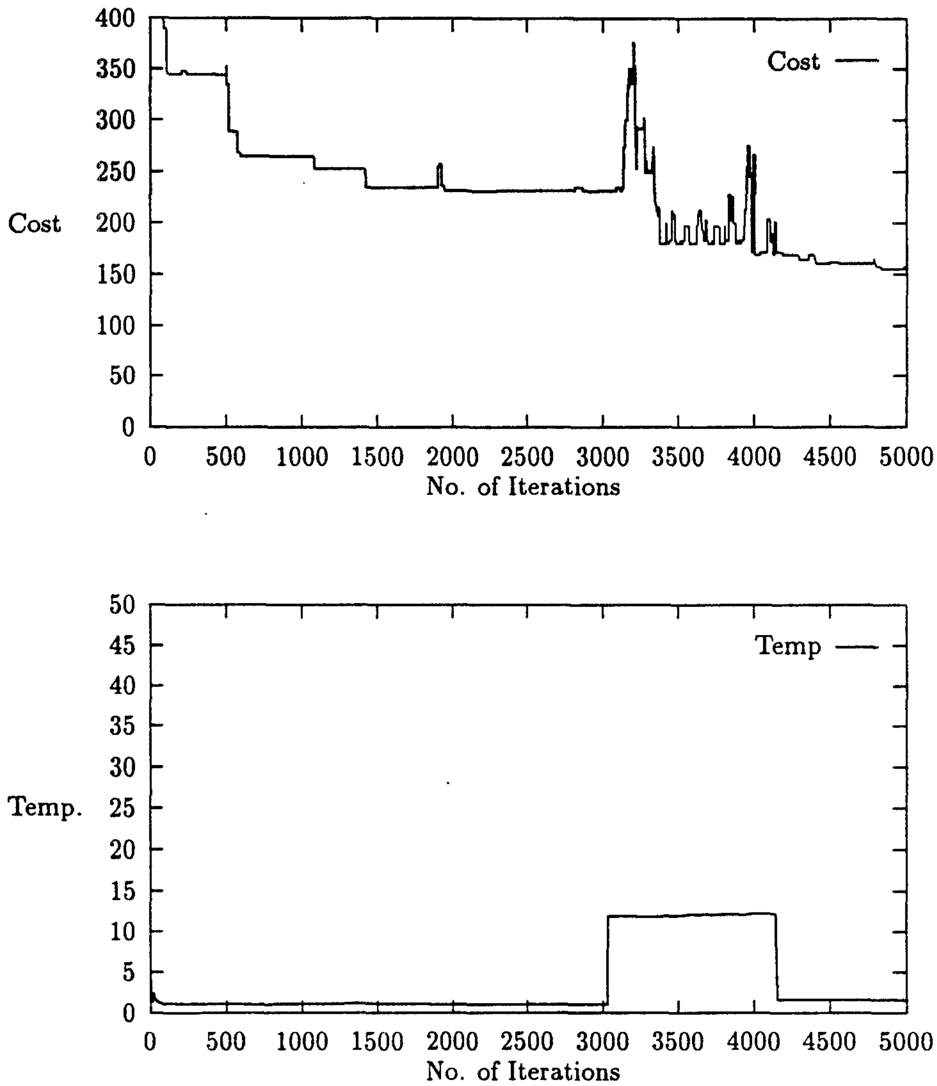

Figure 16: Cost and Temperature trace for Optimization (5000 iterations)

First Run

```
OP# 85 of Batch 3 done on M/C-1 at time  20: 7:28(10/12/92)
OP# 96 of Batch 6 done on M/C-1 at time  22:56: 0(10/12/92)
OP# 81 of Batch 2 done on M/C-1 at time   3:33:52(10/13/92)
OP# 86 of Batch 3 done on M/C-4 at time   6:52:48(10/13/92)
OP# 113 of Batch 11 done on M/C-1 at time  8:43:12(10/13/92)
FAILURE TYPE 2 on MACHINE M/C-1 at time   8:43:12(10/13/92)
 RESCHEDULE.....  HALTING SIMULATION
```

Second Run

```
OP# 97 of Batch 6 done on M/C-4 at time  16: 3:44(10/13/92)
OP# 82 of Batch 2 done on M/C-2 at time  17:53:36(10/13/92)
OP# 114 of Batch 11 done on M/C-3 at time 18:32:32(10/13/92)
OP# 87 of Batch 3 done on M/C-2 at time  23:37:36(10/13/92)
OP# 83 of Batch 2 done on M/C-4 at time   4:59:12(10/14/92)
OP# 100 of Batch 7 done on M/C-1 at time 10:44:48(10/14/92)
OP# 98 of Batch 6 done on M/C-2 at time  12:48:32(10/14/92)
OP# 93 of Batch 5 done on M/C-1 at time  17:57:20(10/14/92)
OP# 115 of Batch 11 done on M/C-4 at time 19:31:44(10/14/92)
OP# 101 of Batch 7 done on M/C-4 at time 23: 2:24(10/14/92)
FAILURE TYPE 3 on MACHINE M/C-4 at time  23: 2:24(10/14/92)
 RESCHEDULE.....  HALTING SIMULATION
```

Figure 17: Partial Simulation Run

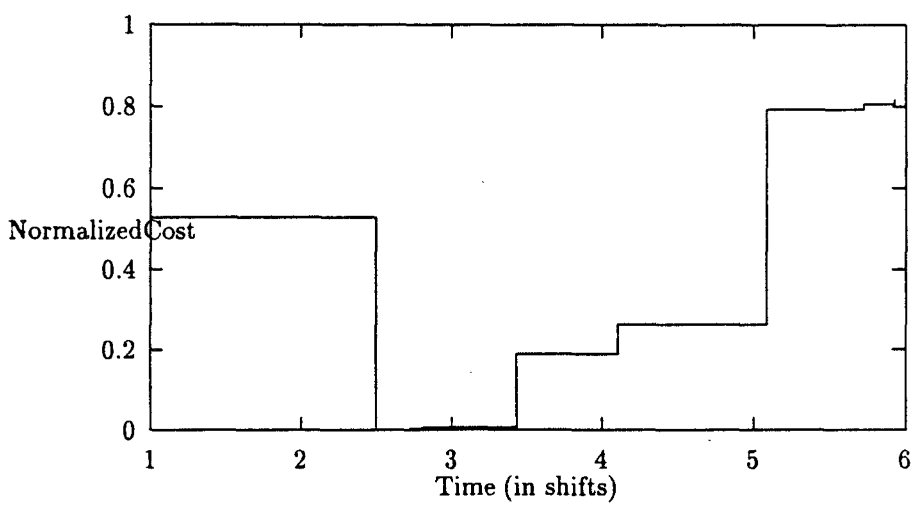

Figure 18: Rescheduling Trace

```
*******************************************************************
*   Batch No.  2          (QTY =  350)                            *
*  mach   setup        start time      finish time    proc time(hr)*
*******************************************************************
*  M/C-1   120    0:56: 0(10/13/92)    3:33:52(10/13/92)    2.63   *
*  Sched times    0: 9:36(10/13/92)    2:56: 0(10/13/92)    2.78   *
*******************************************************************


*******************************************************************
*   Batch No.  3          (QTY =  200)                            *
*  mach   setup        start time      finish time    proc time(hr)*
*******************************************************************
*  M/C-1    45   19:22:40(10/12/92)   20: 7:28(10/12/92)    0.75   *
*  Sched times   19:22:40(10/12/92)   20:10:40(10/12/92)    0.79   *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -*
*  M/C-4    30    3:22:40(10/13/92)    6:52:48(10/13/92)    3.50   *
*  Sched times    3:22:40(10/13/92)    6:57: 4(10/13/92)    3.57   *
*******************************************************************


*******************************************************************
*   Batch No.  6          (QTY =  500)                            *
*  mach   setup        start time      finish time    proc time(hr)*
*******************************************************************
*  M/C-1    45   20:52:16(10/12/92)   22:56: 0(10/12/92)    2.06   *
*  Sched times   20:10:40(10/12/92)   22: 9:36(10/12/92)    1.98   *
*******************************************************************


*******************************************************************
*   Batch No. 11          (QTY =  400)                            *
*  mach   setup        start time      finish time    proc time(hr)*
*******************************************************************
*  M/C-1   120    5:33:52(10/13/92)    8:43:12(10/13/92)    3.15   *
*  Sched times    2:56: 0(10/13/92)    6: 6:56(10/13/92)    3.17   *
*******************************************************************
```

Figure 19: Schedule Timings for Completed Operations (First Run)

```
*******************************************************************
*   Batch No.  2          (QTY =  350)                           *
*  mach    setup         start time      finish time     proc time(hr)*
*******************************************************************
*  M/C-2    20     9: 2:56(10/13/92)   17:53:36(10/13/92)    8.84   *
*  Sched times    9: 3:28(10/13/92)   17:23:12(10/13/92)    8.33   *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -*
*  M/C-4    45   17: 4: 0(10/13/92)    4:59:12(10/14/92)   11.92   *
*  Sched times   17:16:16(10/13/92)    5:46: 8(10/14/92)   12.50   *
*******************************************************************
*******************************************************************
*   Batch No.  3          (QTY =  200)                           *
*  mach    setup         start time      finish time     proc time(hr)*
*******************************************************************
*  M/C-2    60   18:53:52(10/13/92)   23:37:36(10/13/92)    4.73   *
*  Sched times   18:23:28(10/13/92)   23: 8:48(10/13/92)    4.76   *
*******************************************************************
*******************************************************************
*   Batch No.  5          (QTY =  350)                           *
*  mach    setup         start time      finish time     proc time(hr)*
*******************************************************************
*  M/C-1   150   13:14:40(10/14/92)   17:57:20(10/14/92)    4.71   *
*  Sched times   13:12:32(10/14/92)   17:22:40(10/14/92)    4.17   *
*******************************************************************
*******************************************************************
*   Batch No.  6          (QTY =  500)                           *
*  mach    setup         start time      finish time     proc time(hr)*
*******************************************************************
*  M/C-4     0     8:43:12(10/13/92)   16: 3:44(10/13/92)    7.34   *
*  Sched times     8:43:12(10/13/92)   16:31:28(10/13/92)    7.80   *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -*
*  M/C-2    60    0:37:20(10/14/92)   12:48:32(10/14/92)   12.19   *
*  Sched times   23: 8:48(10/13/92)   11: 3:28(10/14/92)   11.90   *
*******************************************************************
*******************************************************************
*   Batch No.  7          (QTY =  250)                           *
*  mach    setup         start time      finish time     proc time(hr)*
*******************************************************************
*  M/C-1    30     9:13: 4(10/14/92)   10:44:48(10/14/92)    1.53   *
*  Sched times     9:13: 4(10/14/92)   10:42:40(10/14/92)    1.49   *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -*
*  M/C-4    30   20: 1:36(10/14/92)   23: 2:24(10/14/92)    3.01   *
*  Sched times   20:33:36(10/14/92)   23:31:44(10/14/92)    2.98   *
*******************************************************************
*******************************************************************
*   Batch No. 11          (QTY =  400)                           *
*  mach    setup         start time      finish time     proc time(hr)*
*******************************************************************
*  M/C-3    20     9: 2:56(10/13/92)   18:32:32(10/13/92)    9.50   *
*  Sched times     9: 3:28(10/13/92)   18:34:40(10/13/92)    9.52   *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -*
*  M/C-4    45     5:44: 0(10/14/92)   19:31:44(10/14/92)   13.79   *
*  Sched times     5:46: 8(10/14/92)   20: 3:12(10/14/92)   14.29   *
*******************************************************************
```

Figure 20: Schedule Timings for Completed Operations (Second Run)

****
************************** PCB 01 ***************************************
****

Operation ID	Machine ID	Setup Time	Batch Qty	Start Time	Finish Time
Batch ID: 007					
0099	(parts)	0	250		08:43:12 (10/13)
0100	M/C-1	30	250	09:13:12 (10/14)	10:42:29 (10/14)
0101	M/C-4	30	250	20:33:23 (10/14)	23:31:57 (10/14)
				DUE:	23:00:00 (10/14)
Batch ID: 000					
0074	(parts)	0	750		08:43:12 (10/13)
0075	M/C-1	30	750	23:49:38 (10/14)	04:17:29 (10/15)
0076	M/C-4	30	750	07:49:38 (10/15)	16:45:21 (10/15)
				DUE:	23:00:00 (10/15)
Batch ID: 009					
0105	(parts)	0	500		08:43:12 (10/13)
0106	M/C-1	30	500	19:27:58 (10/15)	22:26:32 (10/15)
0107	M/C-4	30	500	15:30:21 (10/16)	21:27:29 (10/16)
				DUE:	23:00:00 (10/16)

****
************************** PCB 02 ***************************************
****

Operation ID	Machine ID	Setup Time	Batch Qty	Start Time	Finish Time
Batch ID: 003					
0087	M/C-2	60	200	18:23:12 (10/13)	23:08:55 (10/13)
				DUE:	23:00:00 (10/12)
Batch ID: 006					
0097	M/C-4	0	500	08:43:12 (10/13)	16:31:14 (10/13)
0098	M/C-2	60	500	23:08:55 (10/13)	11:03:12 (10/14)
				DUE:	23:00:00 (10/14)
Batch ID: 010					
0108	(parts)	0	200		08:43:12 (10/13)
0109	M/C-1	45	200	10:36:46 (10/15)	11:24:23 (10/15)
0110	M/C-4	30	200	18:36:46 (10/15)	22:11:03 (10/15)
0111	M/C-2	60	200	02:36:46 (10/16)	07:22:29 (10/16)
				DUE:	23:00:00 (10/16)

Figure 21: Optimal Schedule after Rescheduling. (Part I)

```
*****************************************************************************
************************** PCB 03 *******************************************
*****************************************************************************
Operation Machine  Setup  Batch  Start                  Finish
   ID        ID    Time    Qty   Time                   Time
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
Batch ID: 002
 0082   M/C-2       20     350   09:03:12 (10/13)      17:23:12 (10/13)
 0083   M/C-4       45     350   17:16:14 (10/13)      05:46:14 (10/14)
                                             DUE:      23:00:00 (10/13)
Batch ID: 011
 0114   M/C-3       20     400   09:03:12 (10/13)      18:34:38 (10/13)
 0115   M/C-4       45     400   05:46:14 (10/14)      20:03:23 (10/14)
                                             DUE:      23:00:00 (10/14)
Batch ID: 004
 0088       (parts)  0     450                         08:43:12 (10/13)
 0089   M/C-1      120     450   06:17:29 (10/15)      09:51:46 (10/15)
 0090   M/C-2       20     450   14:17:29 (10/15)      01:00:21 (10/16)
 0091   M/C-4       45     450   22:56:03 (10/15)      15:00:21 (10/16)
                                             DUE:      23:00:00 (10/15)


*****************************************************************************
************************** PCB 04 *******************************************
*****************************************************************************
Operation Machine  Setup  Batch  Start                  Finish
   ID        ID    Time    Qty   Time                   Time
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
Batch ID: 005
 0092       (parts)  0     350                         08:43:12 (10/13)
 0093   M/C-1      150     350   13:12:29 (10/14)      17:22:29 (10/14)
 0094   M/C-3      120     350   21:12:29 (10/14)      09:42:29 (10/15)
                                             DUE:      23:00:00 (10/14)
Batch ID: 008
 0102       (parts)  0     500                         08:43:12 (10/13)
 0103   M/C-1      150     500   17:22:29 (10/14)      23:19:38 (10/14)
 0104   M/C-3      120     500   09:42:29 (10/15)      03:33:55 (10/16)
                                             DUE:      23:00:00 (10/15)
Batch ID: 001
 0077       (parts)  0     425                         08:43:12 (10/13)
 0078   M/C-1      150     425   13:54:23 (10/15)      18:57:58 (10/15)
 0079   M/C-3      120     425   03:33:55 (10/16)      18:44:38 (10/16)
                                             DUE:      23:00:00 (10/16)
*****************************************************************************
*****************************************************************************

Active Time of schedule: 08:43:12, Oct 13, 1992
```

Figure 22: Optimal Schedule after Rescheduling. (Part II)

$qty : \mathcal{B} \rightarrow Z^+$	$qty(b)$ is the quantity of batch b
$earliest\_start : \mathcal{B} \rightarrow \Re$	Batch b cannot start before $earliest\_start(b)$.
$due\_time : \mathcal{B} \rightarrow \Re$	Batch b is due at $due\_time(b)$.
$priority : \mathcal{B} \rightarrow \mathcal{W}$	This is used by the cost function to be defined later.
$proc\_time : \mathcal{O} \rightarrow \Re^+$	Processing time per part for an operation.
$principal\_setup\_time : \mathcal{O} \rightarrow \Re^+$	The time required to setup the machine for a given operation.
$setup\_time : \mathcal{OI} \times \mathcal{OI} \rightarrow \Re^+$	Setup time between two operations
$transfer\_delay : \mathcal{O} \rightarrow \{\text{BATCH}\} \bigcup \Re^+$	The transit delay from the start of an operation to the start of the next.
$transfer\_time : \mathcal{OI} \rightarrow \Re^+$	Delay in start of operation
$method\_family : \mathcal{P} \rightarrow \mathcal{MD}$	$\mathcal{MD}$ is the set of Method family

Derived Set definitions

$\mathcal{OI} \equiv \{(b, o) | b \in \mathcal{B} \bigwedge o \ \underline{in} \ op\_seq(part(b))\}$

$\mathcal{SOI} \equiv \{< k_1, k_2, ..., k_n > | k_i \neq k_j \text{ when } i \neq j \bigwedge k_i \in \mathcal{OI} \text{ for all } i\}$

These are *operation instances* and *sequences of operation instances*, respectively.

Appendix B

Transfer time calculation

algorithm $tranfer\_time((b, o))$
    if $transfer\_delay((b, o)$ =BATCH
        return $(qty(b) \times proc\_time(o))$
    else
        return $transfer\_delay((b, o))$
    endif
end_algorithm

Setup time calculation

algorithm $setup\_time((b_1, o_1), (b_2, o_2))$
    $part_i = part(b_i)$, for $i = 1, 2$.

if $method\_family(part_1) = method\_family(part_2)$
    return β $principal\_setup\_time(o_2)$ /* for some β, $0 < \beta < 1$ */
else
    return $principal\_setup\_time(o_2)$
endif
end_algorithm

Start and finish time calculation

if $\{(b_2, o_2) \in \mathcal{OI} | ((b_2, o_2), (b, o)) \in \mathbf{E_s}\} = \emptyset$
    $t_s(b, o) = earliest\_start((b, o))$
else
    $t_s((b, o)) =$

$$\max_{\{(b_2,o_2)|((b_2,o_2),(b,o))\in \mathbf{E_S}\}} \begin{cases} t_s((b_2, o_2)) + transfer\_time((b_2, o_2)) & \text{if } b_2 = b \\ t_f((b_2, o_2)) + setup\_time((b_2, o_2), (b, o)) & \text{otherwise} \end{cases}$$

$$t_f((b, o)) = \max\{t_s((b, o)) + qty(b) \times proc\_time(o), \max_{\{(b,o_2)\in \mathcal{OI} | o \underline{\text{ follows }} o_2 \underline{\text{ in }} op\_seq(part(b))\}}(t_f(b, o_2))\}$$

<u>Appendix C</u>

```
algorithm INSERT_NEW_BATCH

   while (some operation in new batch is not in schedule)

      OP = next unscheduled operation in new batch;

      if (OP can be inserted without
                increasing the schedule cost)
         put OP in schedule at zero cost;
      else
         put OP in schedule at minimum cost;
      endif

   end_while

end_algorithm INSERT_NEW_BATCH
```

References

[1] Jastej S. Dhingra. *Reconfigurable Control in Discrete Event Dynamic Systems applied to Manufacturing Systems.* PhD thesis, University of Maryland, College Park, 1992.

[2] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1091, 1953.

[3] D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli. Convergence and finite-time behavior of simulated annealing. In *Proc. 24th IEEE Conf. on Decision and Control*, pages 761–767, 1985.

[4] K. L. Musser, J. S. Dhingra, and G.L. Blankenship. Optimization based job shop scheduling. Technical Report TR-91-9, Systems Research Center, University of Maryland, 1991. To appear in IEEE Transactions on Automatic Control.

[5] Keith L. Musser. *A Distributed Algorithm for Scheduling in Discrete Manufacturing.* PhD thesis, University of Maryland, College Park, 1992.

[6] E.H.L. Aarts P.J.M. van Laarhoven and J.K. Lenstra. Job shop scheduling by simulated annealing. *Operations Research*, 40(1):113–125, Jan.-Feb. 1992.