MATCH-UP REAL-TIME SCHEDULING

James C. Bean
John R. Birge

Department of Industrial & Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109

June 1985

Technical Report 85-22

MATCH-UP REAL-TIME SCHEDULING*

James C. Bean
John R. Birge

Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, Michigan 48109

ABSTRACT

Scheduling algorithms are often of limited benefit in practical situations because of disruptions that change the problem during operation. One approach to this changing environment is to reschedule so that the schedule returns to the pre-planned schedule. This paper presents an analytic justification for this heuristic. We show that an optimal recovery from a single disruption does indeed return to the pre-schedule under certain conditions. We also give bounds on the error that may result from using the heuristic with a predetermined time to match-up to the pre-schedule. These bounds are extended to an infinite horizon stochastic scheduling problem with multiple disruptions following some distributional assumptions.

1. Introduction

A vast majority of the literature dealing with production scheduling involves determining a good schedule over a substantial time frame assuming that all problem characteristics are known. Such schedules have two uses. As a planning tool, they can be used to order material, set work schedules and other functions preliminary to the actual production. As an operational· device, they can be used to direct step-by-step production operations.

In this second capacity such deterministic scheduling procedures typically run into difficulty. Once the production process begins, random disruptions can force the system out of the prescribed states rendering the preformulated schedule invalid. Such disruptions can include delays in the

* This material is based upon work supported by the National Science Foundation under Grants Nos. ECS-8304065 and ECS-8409682.

arrival of materials or components, quality rejection of material or components, machine breakdown or operator absences.

We would like to anticipate such disruptions during the pre-scheduling of the system and build a schedule with recourse for each contingency. Research such as Pinedo and Ross [1980], Glazebrook [1981], and Pinedo [1983] is currently improving capabilities in these directions. At this time, the available techniques are not able to solve problems of the size and complexity needed to make operational contributions to actual production systems.

In the absence of a tractable optimal technique most scheduling practitioners use control type real-time heuristics. Most common are priority list scheduling rules. In such techniques available jobs are ordered by some simple rule. When a machine becomes available, the job heading the list is started. These techniques are myopic and can lead to substantial error.

This paper develops the framework for an alternative approach to the disruption problem that yields operational heuristic scheduling systems. In this approach we seek to use the information captured in the deterministic pre-schedule, while altering its details to compensate for disruptions. Such alterations must be done in real time as the disruptions occur. In order to react in real time while retaining the "goodness" of the schedule, we do not completely reschedule tasks, but rather, adapt the old schedule to smooth out the difficulties created by the disruption and match-up with the pre-schedule. This approach has intuitive appeal since material flows have been set according to this pre-schedule. We show that the optimal schedule, given the disruption, attempts to match up to the pre-schedule. Hence, by attempting to match-up in our heuristic we are moving in the appropriate direction.

In this paper we propose a dynamic optimization formulation of a general scheduling problem based on problems faced at a large auto manufacturer. This allows the use of economic turnpike theory (see McKenzie [1976]) as a foundation for adaptive approaches to real-time scheduling. Turnpike results are asymptotic. In other words, adaptive approaches are justified in the limit as the horizon of the problem is increased. Implementation of match-up procedures requires choosing a finite time where the schedule is expected to match up with the predetermined schedule. The imposition of this finite match-up time may cause error. However, bounds may be developed for these errors.

Section 2 includes a formal model of the problem class being addressed and the procedure proposed here. Section 3 contains a theoretical justification of the procedure based on turnpike theory. Sections 4 and 5 include a discussion of the implementation of adaptive scheduling and a derivation of error bounds. Finally, Section 6 contains a summary and conclusions.

2. Problem Statement

Though most formulations of scheduling problems deal with a finite set of jobs, implying a finite problem horizon, actual scheduling problems have indefinite horizons. Though a finite set of jobs is known at any point in time, as those jobs are completed new jobs are defined so that the process continues indefinitely. The assumption that we may truncate the problem to currently known jobs may not be valid due to the introduction of end–of–study effects (see Baker [1977]).

In this section we view the scheduling problem as a deterministic infinite horizon optimization problem. Over the infinite horizon an infinite number of jobs will be defined and completed, but we assume that at any finite time there are only a finite number of jobs to consider. Let n_k be the maximum number of jobs at any finite time k.

The scheduling system is modeled as a sequence of decisions detailing task processing during the next time period. The system can be viewed as being in one of many states, each of which details the accumulated processing of known jobs. Resource assignments may also be included in state definition without difficulty but are omitted here to simplify notation. Constraints on the system such as task precedence, preemptability, and resource capacities can be seen as restrictions on the transition of the system from one state to another.

2.1 Notation

For completeness a list of all notation used in the paper is included here. The need for some of the following will become clear later in the paper.

$n_k \in Z$: The number of jobs in the system at time k.

$N \in Z$: A finite horizon time.

$x_k \in \Re^{(n_k)}$: The state of the system at the end of processing in period k. The i^{th} element of x_k, x_k^i, is the accumulated processing of job i.

x_k^*: The state passed through at time k by an undisrupted weakly optimal schedule.

X_k^*: The set of states corresponding to any undisrupted weakly optimal schedule.

$x_k^*(\bar{x})$: The state passed through at time k by a weakly optimal schedule that begins at state $\bar{x}$ at time 0.

$\pi_k \in [0,1]^{n_k}$: A decision vector indicating the fraction of time each job is processed during period k.

$\pi \in \times_{k=1}^{\infty}[0,1]^{n_k}$: Decision vectors describing a schedule covering the infinite horizon.

π^*: A weakly optimal schedule.

$\pi^*(\bar{x})$: A weakly optimal schedule given that the system is in state $\bar{x}$ at time 0.

Π: The set of all feasible schedules, π.

$\phi_k(\pi) \in \Re$: The incremental cost from schedule π during period k.

$\Phi(\pi; N) \in \Re$: The cumulative cost of schedule π over the first N periods, $= \sum_{k=1}^{N} \phi_k(\pi)$.

$\Phi(\pi) \in \Re$: The cumulative infinite horizon cost $= \lim_{N \to \infty} \Phi(\pi; N)$.

$\Phi'(a, b) \in \Re$: The minimal cost of moving from state a to state b, where the periods for the states are not specified.

$\Phi^*(N) \in \Re$: The optimal cost over the first N periods, where any disruption restrictions are included.

$\hat{\Phi}^* \in \Re$: The optimal infinite horizon cost after a single disruption.

$\Phi^* \in \Re$: The optimal infinite horizon cost from $k = 0$ including disruptions.

$\bar{\Phi}^* \in \Re$: The optimal expected infinite horizon cost from $k = 0$ including disruptions.

$D_k \in \Re^{(n_{k-1})} \times \Re^{(n_k)}$: The set of feasible transitions at time k. That is, given the current state, x_{k-1}, and a potential next state, x_k, then $(x_{k-1}, x_k) \in D_k$ if and only if (x_{k-1}, x_k) represents a feasible transition. The set of feasible π_k, given the current state, can be inferred from D_k.

p_i: The processing time required for job i.

w_i: The weight assigned job i (to be used for weighted tardiness or weighted flowtime schedules).

$\bar{k}_i$: The time for matching up after the ith disruption.

$\underline{k}_i$: The time of the ith disruption.

$\bar{l}$: The time between a disruption and a match-up.

δ: The cost of following the match-up heuristic from a single disruption until return to the pre-schedule.

δ_i: The cost of following the match-up heuristic from $\underline{k}_i$ to $\bar{k}_i$.

d: The minimum cost from a single disruption state to any potentially optimal state at $\bar{l}$.

P_k: The set of potentially optimal states at period k.

d_i^1: The minimum cost from any state in $P_{\underline{k}_i}$ to any state in $P_{\bar{k}_i}$.

d_i^2: The minimum cost from any state in $P_{\bar{k}_i}$ to any state in $P_{\underline{k}_{i+1}}$.

2.2 Assumptions

A1) Total costs are the sum of incremental costs in each period.

A2) The incremental costs in each period are convex in the processing decisions.

A3) The objective is to minimize total costs.

A4) In weighted flowtime and tardiness problems there is a positive lower bound on w_i.

A5) There is sufficient slack time in the infinite horizon schedule that for any feasible initial state, $\bar{x}$, and schedule, π, there exists a feasible schedule, $\pi(\bar{x})$, and time $\bar{l}$, uniform over $\bar{x}$, such that for $k \geq \bar{l}$, $\pi_k = \pi_k(\bar{x})$. Further, the absolute cost to achieve this matching up, δ, is bounded above by U, uniform in $\bar{x}$. That is, any feasible schedule is reachable frcm any feasible initial state at reasonable cost.

A6) Several technical regularity conditions which are discussed in the appendix.

2.3 Infinite Horizon Objective

In infinite horizon optimization problems there are many ways to define the infinite horizon objective. If the infinite horizon costs converge, the optimal schedule is that with least cost. However, such convergence can usually be assumed only under sufficient discounting. Since scheduling problems are rarely discounted, all schedules will possibly have infinite cost.

Several definitions of optimality have been suggested for this case, including: average optimality (see Derman [1966]), 1-optimality (see Blackwell [1962]), overtaking cptimality (see McKenzie), catching-up optimality (see McKenzie), forecast horizon optimality, and periodic forecast horizon optimality (see Hopp, Bean and Smith [1984]). To conform with the concepts of McKenzie, we use the concept of a weakly optimal schedule in the first part of the paper. As will be proven, weakly optimal implies average optimal.

We say that a schedule, π, *overtakes* another schedule, $\bar{\pi}$, if

$$\liminf_{N\to\infty} \sum_{k=1}^{N} [\phi_k(\pi) - \phi_k(\bar{\pi})] \geq \epsilon$$

for some $\epsilon > 0$. A schedule is *weakly optimal* if no other schedule overtakes it. We assume here that such a schedule exists. Given such a schedule, redefine $\phi_k(\pi)$ by subtracting the incremental cost of the weakly optimal schedule. Then, without loss of generality, the optimal cost is zero. We make the additional assumption:

A7) A weakly optimal schedule exists and its incremental costs have been used to normalize the cost function such that the infinite horizon optimal undisrupted cost is zero over all time intervals.

A schedule, $\hat{\pi}$, is said to be *average optimal* if

$$\liminf_{N\to\infty} \{\Phi(\hat{\pi}, N)/N - \Phi(\pi, N)/N\} \geq 0, \quad \forall \pi \in \Pi.$$

Lemma 1: *If π^* is weakly optimal then it is average optimal.*

Proof: In the definition of weakly optimal divide both sides by N. The result follows.∎

The term *cost* used here represents any of several of the common scheduling objectives, as well as any more general objective that satisfies the stated assumptions. The objective primarily used for examples in this paper is weighted tardiness. Assumption A2 requires that the incremental costs be convex. This assumption is satisfied for most common scheduling objectives including weighted flow time and weighted tardiness.

Lemma 2: *If $w_i \geq 0$ for all i, then incremental weighted tardiness in period k is convex in processing time.*

Proof: As the processing for a particular job is increased, the tardiness of the job (if any) decreases linearly with slope $-w_i$ until there is no further tardiness. At that time it remains at zero regardless of further processing. The resulting curve is convex in processing of job i.■

It is equally simple to show that weighted flowtime is convex in processing for non-negative weights. Note that since the sum of convex functions is convex, the accumulated cost through time N is also convex for either of these criteria.

2.4 Feasibility

All feasibility conditions are assumed described by the sets of conditionally feasible decisions, D_k. Common constraints include non-preemptability, precedence, resource capacities, and maximum allowable tardiness. Each of these can be represented by restrictions which retain the necessary characteristics of D_k. For example, if a job is non-preemptable and its current processing is strictly between zero and its total processing time, the only feasible choice is to process until processing is completed. For precedence, processing a subsequent job would require that total processing on its predecessor equals its necessary processing.

3. Turnpike Results

The model developed in Section 2 can be viewed as an investment model similar to that described in Ramsey [1928]. McKenzie shows that under certain assumptions such models behave in a manner analogous to the matching-up described here. In the literature of economics and infinite horizon optimization these results are called turnpike theorems. The analogy is from shortest path problems on a surface road and turnpike network.

Clearly, the shortest path from one point on a turnpike to another on the same turnpike is that turnpike. Even if you live off the turnpike, if the destination is distant enough, the shortest path is to take surface streets to the turnpike and to take the turnpike from there. In our model the turnpike is analogous to the predetermined schedule (see Figure 1). The distance from the current state to the turnpike exists because some disruption has transferred the problem to an alternate state.

If these results can be shown to apply to a general form of the scheduling problem they have significant implications for appropriate heuristic approaches. After a disruption, the optimal new schedule would then converge to the optimal schedule derived before processing began. This pre–

schedule is then a surrogate for the complex objectives involved in scheduling. If our heuristic is designed to return to the original schedule, we know that this is in fact the direction followed by the optimal schedule to the disrupted problem.

The strongest turnpike results require uniform convexity of the incremental costs. Though most measures used in scheduling problems do not satisfy this assumption, the stronger results are valuable for perspective and potential application to other objectives.

3.1 The Uniform Convexity Case

If the incremental cost functions are uniformly convex, as the system proceeds in time from the boundary condition (disruption), the sequence of states passed through by the system, and hence the decisions made, grow increasingly close to the states defined by the pre–schedule. The effect of the disruption diminishes.

The hypothesis of this theorem requires the concept of a uniformly convex function. Essentially, to be uniformly convex a function must be strictly convex and must not asymptotically approach a line. For a rigorous definition of this notion see McKenzie.

Theorem 3: *Given the assumptions of this paper, if $\phi_k(\pi)$ is uniformly conver for all k and π corresponding to D_k, then $x_k^*(\bar{x}) \rightarrow x_k^*$ as $k \rightarrow \infty$.*

Proof: This problem seeks to minimize the sum of uniformly convex functions. This is equivalent to maximizing the negatives of these functions. The negatives of uniformly convex functions are uniformly concave. The sum retains this characteristic. Hence, the conditions of Theorem 3 of McKenzie are satisfied. The conclusion of this theorem follows immediately.∎

The proof in McKenzie follows this line. The cost of passing through states x_k^* is zero by Assumption A7. There exists a feasible path from $\bar{x}$ which matches up with this schedule. By Assumption A5 this may be done by time $\bar{l}$ at a cost of no more than U. Hence, the cost of passing through $\{x_k^*(\bar{x})\}$ given an initial state of $\bar{x}$ must be better that U. By uniform convexity, if $\bar{x}_k^*$ is bounded away from x_k^*, then a uniform penalty will be charged each period. This leads to infinite cost and a contradiction. For further details see McKenzie.

Corollary 4: *Under the conditions of Theorem 3, $\pi_k^*(\bar{x}) \rightarrow \pi_k^*$ as $k \rightarrow \infty$.*

Proof: In general, $\pi_k = x_k - x_{k-1}$. The result follows immediately from Theorem 3.

3.2 The Case of Multiple Optima

When the incremental cost functions are not uniformly convex, the problem may have multiple optima. This is the case in most scheduling objectives, including weighted flowtime and tardiness. If there are multiple optima, different initial states could lead to weakly optimal schedules which converge to different turnpikes, each being optimal. The claim can still be made that as $k \to \infty$, the state of the disrupted system will approach the set of weakly optimal schedules.

Definition: $\rho(x_k, X_k^*) = \inf_{y \in X_k^*} ||x_k - y||$.

Theorem 5: *Under the assumptions of this paper,* $\lim_{k\to\infty} \rho(x_k, X_k^*) = 0$.

Proof: Follows directly from Section 7 of McKenzie.

3.3 Example

Consider the following single machine total tardiness problem. At the beginning of each seven day cycle four jobs become ready. Their total processing is six days. Their due dates are such that they can be completed without tardiness—provided there are no disruptions in the system. The optimal schedule over the infinite horizon is to sequence the jobs 1–2–3–4 each week. This schedule is weakly optimal as discussed above.

Assume that the machine is now down for the first four days of the upcoming week. This disruption will not only affect the schedule for this week, but for upcoming weeks as well. The consequence of the theorem above is that we know that the effects of this disruption will fade out and that the weekly schedule will approach the original weakly optimal schedule.

Data for the example appear in Table 1. Figure 2 displays the undisrupted, weakly optimal schedule and the optimal schedule given the disruptions. Note that after four weeks the two schedules are identical for all future periods.

4. Implementation

Theorems 3 and 5 cannot be used directly to control a scheduling system in real time. They do, however, suggest that the following heuristic may be efficient. Assume that a schedule has been determined and used to set material flows. When a disruption occurs, the pre–schedule becomes obsolete. Rather than rescheduling in real time using the original objective, substitute as the objective a desire to return at minimal cost to the pre–schedule at some future date. The pre–schedule is assumed to incorporate all important goodness characteristics. It should contain more of these characteristics than could be incorporated explicitly in real time. From Theorems 3 and 5 we know that this heuristic schedule is tracking in the same direction as the optimal schedule, given the initial state.

The implemented process contains the following steps.

Match–Up Heuristic

Step 0: Construct a pre–schedule.

When a disruption occurs

Step 1: Determine some future time, $\bar{l}$, where the pre-schedule is reachable from the current state.

Step 2: Reschedule from current time 0 to $\bar{l}$ beginning in the current state and ending in state $x_{\bar{l}}^*$ at time $\bar{l}$. Stop.

This heuristic does not guarantee an optimal solution in all situations. However, by Theorems 3 and 5 we know that $\bar{l}$ can be chosen large enough to get arbitrarily close to an optimal schedule. It is not, however, generally known how large $\bar{l}$ must be or when disruptions will occur. A large $\bar{l}$ may also lead to computational difficulties. making the method impractical. Given these potential problems, the algorithm must be implemented as a heuristic and $\bar{l}$ must be determined to balance error and effort. Methods for bounding the error are given in the following section.

5. Error Bounds

The problem formulation presented here represents too general a class of scheduling problems to derive tight bounds on the error caused by this class of heuristics. For a specific problem, bounds should be derived to help set $\bar{l}$. In the following sections we present a rough bound for the general problem and apply it to an example.

5.1 Single Disruption

We obtain bounds on the optimal schedule cost by comparing our heuristic with a myopic solution strategy. Sharper bounds exploiting problem structure may be obtained using the aggregation procedure in Bean, Birge, and Smith [1984].

For a single disruption, we assume that the match-up heuristic for some disruption state returns to state $x_{\bar{l}}^*$. The cost of returning to $x_{\bar{l}}^*$ is assumed to be δ.

Definition: The set of *potentially optimal* states at some time k, P_k, is defined as the subset of states, feasible after the disruption, which may be on an optimal path. In the absence of information, this can be taken to be the set of states feasible after the disruption. Note that this involves two reductions: elimination of states no longer feasible, and elimination of states that can be proven not to be on any optimal path.

Let the optimal cost of continuation after disruption be $\hat{\Phi}^*$. A myopic solution strategy is to find a path with minimum cost d from the disruption state to any state in $P_{\bar{l}}$.

Lemma 5: *Given the assumptions of this paper, the single disruption optimal schedule cost $\hat{\Phi}^*$ is bounded by*

$$d \leq \hat{\Phi}^* \leq \delta.$$

Proof: An optimal path must pass from the disruption state to some potentially optimal state at $\bar{l}$.

The value d is the minimum among these distances, giving the lower bound. The upper bound follows from Assumption A7.■

5.2 Multiple Disruptions

In general, the system will encounter more than one disruption. For this possibility, we again use a myopic approach and the match-up heuristic to obtain bounds on the optimal solution cost. We let $\bar{l}$ be the time between the ith disruption at $\underline{k}_i$ and the ith match-up time $\bar{k}_i$ (i.e., $\bar{k}_i = \underline{k}_i + \bar{l}$) for all i. We assume here :

A8) If consecutive disruptions occur at times $\underline{k}_i$ and $\underline{k}_{i+1}$, then all potentially optimal states at $\underline{k}_{i+1}$ are reachable from all potentially optimal states at $\underline{k}_i$.

A9) $\underline{k}_{i+1} \geq \bar{k}_i$ for all i. That is, disruptions are sufficiently infrequent that we can match-up before the next disruption.

These assumptions ensure the feasibility of the match-up heuristic.

Let M be the number of disruptions for some finite horizon problem. The myopic approach is to find the minimum cost schedule path from $P_{\underline{k}_i}$ to $P_{\bar{k}_i}$ and from $P_{\bar{k}_i}$ to $P_{\underline{k}_{i+1}}$. Following this idea leads to a local minimization for lower bounds. Let

$$d_i^1 = \min\{\Phi'(a,b) : a \in P_{\underline{k}_i}, b \in P_{\bar{k}_i}\}, i = 1, \dots, M,$$

$$d_0^2 = \min\{\Phi'(x_0, b) : b \in P_{\underline{k}_1}\},$$

and

$$d_i^2 = \min\{\Phi'(a,b) : a \in P_{\bar{k}_i}, b \in P_{\underline{k}_{i+1}}\}, i = 1, \dots, M-1,$$

where $\Phi'(a,b)$ is the minimum cost over all paths connecting states a and b. These values are lower bounds on the values that the myopic approach could obtain. They also bound the cost of any other path over their defined ranges.

The match-up heuristic finds the minimum cost from the state following the disruption to $x_{\bar{k}_i}^*$ (within the disruption restricted set of feasible paths). Let this cost be δ_i. All other costs are zero for the match-up heuristic by Assumption A7. These observations are contained in the following theorem.

Theorem 6: *Given the assumptions of this paper, the multiple disruption optimal cost Φ^* is bounded by*

$$\sum_{i=1}^{M}(d_i^1 + d_{i-1}^2) \leq \Phi^* \leq \sum_{i=1}^{M} \delta_i,$$

where there are M disruptions.

5.3 Random Disruptions

All of the results above refer to a deterministic environment where all disruptions are known in advance. Instead, we would like to allow for a distribution on the disruptions and then find the optimal expected cost solution. For this stochastic environment, we define a probability space $(\Omega, \mathcal{B}, P)$. Each sample point ω corresponds to a scenario of occurrence times and durations of disruptions. All deterministic quantities defined above are interpreted as random variables when they are indexed by ω. This allows us to consider the stochastic case without defining extensive additicnal notation. The quantity $\Phi(\bar{\pi}, \omega)$ then denotes the infinite horizon cost of following schedule strategy $\bar{\pi}$ under the conditions corresponding to ω. We wish to find

$$\bar{\Phi}^* = \min_{\bar{\pi}} \mathrm{E}(\Phi(\bar{\pi}, \omega)), \tag{1}$$

where $\mathrm{E}(\cdot)$ denotes mathematical expectation. We again assume that the disruptions are distributed so that Assumptions A8 and A9 hold almost surely.

Lemma 7: *Given the assumptions of this paper, the optimal expected cost $\bar{\Phi}^*$ is bounded by*

$$\bar{\Phi}^* \geq \mathrm{E}(\Phi^*(\omega)) \geq \mathrm{E}(\sum_{i=1}^{M(\omega)} (d_i^1(\omega) + d_{i-1}^2(\omega)).$$

Proof: Let $\bar{\pi}^*$ solve (1), then $\Phi^*(\omega) \leq \Phi^*(\bar{\pi}^*, \omega)$. Integration yields the first inequality. The second inequality follows from Theorem 6. ∎

Lemma 7 leads to the following result.

Theorem 8: *Given the assumptions of this paper, the optimal expected cost $\bar{\Phi}^*$ is bounded by*

$$\mathrm{E}(\sum_{i=1}^{M(\omega)} \delta_i(\omega)) \geq \bar{\Phi}^* \geq \mathrm{E}(\sum_{i=1}^{M(\omega)} (d_i^1(\omega) + d_{i-1}^2(\omega)). \tag{2}$$

Proof: The left-hand side of the expression is the expected value of following the match-up strategy. Since it is a feasible strategy, its expected value is an upper bound on the optimal expected cost. The second inequality follows from Lemma 7.∎

The use of these bounds is best demonstrated in an example. In this example, we make assumptions that actually reduce the interval in (2) to a point. For the infinite horizon objective we use expected average cost. In this case, if $\bar{\Phi}^*(N)$ is the optimal expected cumulative disruption cost for an N period horizon, then

$$\bar{\Phi}_{\mathrm{ac}}^* = \lim_{N \to \infty} \frac{\bar{\Phi}^*(N)}{N}.$$

Assuming each disruption causes some loss that is finitely positive, an infinite number of disruptions cause an infinite amount of loss. For this section the weakly optimal definition of optimality is not sufficient. Hence, we use average optimality. This gives us the average loss per period due to disruptions. The single disruption results of Theorems 3 and 5 still apply for this objective since all weakly optimal solutions also minimize average cost (Lemma 1). Matching up with the pre-schedule at some point must, therefore, lead to a minimum average cost solution. The bound in (2) can be applied for any finite horizon N as well as in the limit so that (2) can be used to bound $\bar{\Phi}^*_{ac}$.

The example includes two jobs which are cyclically available and due. Each job has a weight of one. Let Job 1 be ready at times $5t$, have a processing time of three, and be due at times $5t+3$, for $t = 0, 1, 2 \ldots$. Let Job 2 be ready at $5t$, have a processing time of one, and be due at $5t+5$. Then the optimal pre-schedule processes Job 1 at $5t$ and Job 2 at $5t+3$ leaving one unit of slack in each production interval. We assume that disruptions occur with equal probability at 15, 16, 17, 18 or 19 time units since the last disruption and that disruptions last one or two time units with independent equal probabilities.

We choose a match-up time $\bar{l} = 10$ and assume that jobs are resumed after disruption with no loss of processing time. Note that the potentially optimal states at k are $\{x^*_k\}$ for $k(\text{mod } 5) = 0, 1, 2$ or 3 and $x_k = x^*_k$ or $x_k = (3, 0)$ for $k(\text{mod } 5) = 4$. In all cases, $\min\ \{\Phi'(a, b) : a \in P_{\underline{k}_i}, b \in P_{\bar{k}_{i+1}}\} = \Phi'(x^*_{\underline{k}_i}, x^*_{\bar{k}_{i+1}})$. Note also that all costs are non-negative. These two observations imply that, for this problem,

$$\delta_i(\omega) = d^1_i(\omega),$$

and

$$d^2_i(\omega) = 0.$$

From this we obtain

$$\bar{\Phi}^*_N = \mathrm{E}(\sum_{i=1}^{M_N(\omega)} \delta_i(\omega)),$$

and

$$\bar{\Phi}^*_{ac} = \mathrm{E}(\delta_i(\omega)/N_C(\omega)),$$

where M_N is the number of disruptions in the first N periods and N_C is the length of a cycle between disruptions.

For this example the match-up solution produces an optimal average cost infinite horizon solution. The minimum expected average cost can then be calculated by weighting the costs over all cycles. The cycles are found by observing that disruptions occur at 0, 1, 2, 3 or 4 units into a production interval with equal probabilities. Tardiness to the match-up point is then found and weighted by the inverse of the expected time to the next disruption. The resulting values appear in Table 2.

Cyclic availabilities and requirements for jobs are reasonable in many circumstances, but the distribution assumption of this example is restrictive and would not be met in general. The assumptions are however made to show the utility of the bounds. In specific applications, problem structure could be used to generate bounds.

In practice, schedulers can use the bounds in (2) to determine the time to match-up $\bar{l}$. They can consider the possible disruption patterns and calculate recovery costs under different scenarios to find the interval in (2). They can then vary $\bar{l}$ so that this interval and the computational burden for matching up are properly balanced.

6. Summary and Conclusions

Extensive deterministic pre-schedules are often inadequate to operate large systems due to unforeseen system disruptions such as machine failure. A common solution is real-time list processing control type algorithms. We present an alternative here that exploits the global attributes of the pre-schedule. It then adapts by matching up with the pre-schedule after disruption.

We have implemented this match-up heuristic in a computer program developed for a large auto manufacturer. The program considers a system of parallel nonidentical machines. Jobs have ready times and due dates. Each job requires a tool from some finite set of tools and may be completed (with possibly different processing times) on any of its set of compatible machines. A pre-schedule obtained by a global scheduling code is followed until a disruption occurs. The match-up heuristic chooses a subset of the machines to reschedule so that the pre-schedule can again be followed in $\bar{l}$ time units. The objective over this horizon includes weighted tardiness and an incentive to keep jobs on their pre-scheduled machines.

The match-up heuristic code has been applied to a variety of situations using actual data from plants. In ten of eleven actual problems, the heuristic was able to obtain significant improvements over the previously used solution of pushing back the pre-schedule (the eleventh problem tied). The average reduction in tardiness was 36%. Due to the proprietary nature of the results, no further details of testing can be reported at this time. Further testing is planned to assess the heuristic's performance in other situations.

In summary, we have presented a method that adapts for a changing production environment by seeking to match up with a pre-schedule. The method becomes optimal as the rescheduling horizon is lengthened and the interval between disruptions increases. This is shown using results from economic turnpike theory. The error involved in using the method is also bounded by the difference between the match-up cost and a lower bound found by local minimization. Comparisons of the error bounds for different match-up problem sizes can be used to determine the value of additional computational effort in specific situations.

APPENDIX

The following technical assumptions are necessary for the theory in McKenzie. While important to some applications, these are technicalities in the scheduling problems discussed in this paper. They are included for completeness.

Finite Transitions: For any given $\alpha < \infty$, there exists $\beta < \infty$ such that $||x_{k-1}|| < \alpha$ implies that $|\phi_k| < \beta$ and $||x_k|| < \beta$ for all feasible π. For weighted tardiness this is clear since no more than one unit of processing can be done in any period and tardiness is bounded by the weights.

Well-Defined States: As noted, costs over the infinite horizon may diverge. Assume that a weakly optimal schedule is at hand. Alter the cost in each period by subtracting the cost of the weakly optimal schedule in that period. Then, the infinite horizon cost of the weakly optimal schedule is zero. A state at some time k is well–defined if the infinite horizon cost beginning at that state and time is finite using this altered cost structure. Any state not well–defined cannot be continued from in finite cost. Given the assumption of sufficient slack to reach the weakly optimal schedule, the set of well–defined states at time k is $\Re^{(n_k)}$.

Uniform Lower Bound on Convexity of Von Neumann Facets: It is sufficient to assume that there is a uniform positive lower bound on the weights used in weighted tardiness or flowtime. (See McKenzie for further details.)

Uniformly Bounded Von Neumann Facets: This is trivial since no more than one unit of processing can be done in any period. (See McKenzie for further details.)

TABLE 1: Example Problem Data

job #	1	2	3	4
proc. time	1	1	1	3
due time	1	3	5	7

TABLE 2: Expected Average Cost Values

Disruption Position	**Down Time**	**Probability**	**Average Cost**
0	1	0.10	0.059
0	2	0.10	0.237
1	1	0.10	0.059
1	2	0.10	0.237
2	1	0.10	0.059
2	2	0.10	0.237
3	1	0.10	0.000
3	2	0.10	0.118
4	1	0.10	0.000
4	2	0.10	0.059
TOTAL			**0.107**

REFERENCES

Baker, K. [1977], "An Experimental Study of the Effectiveness of Rolling Schedules In Production Planning," **Decision Sciences**, Vol. 8, pp. 19–27.

Bean, J., J. Birge, and R. Smith [1984], "Aggregation in Dynamic Programming," Technical Report 84–10, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan 48109.

Blackwell, D. [1962], "Discrete Dynamic Programming," **Annals of Mathematical Statistics**, Vol. 33, pp. 719–726.

Derman, C. [1966], "Denumerable State Markov Decision Processes—Average Cost Criterion," Annals of Mathematical Statistics, Vol. 37, pp. 1545–1554.

Glazebrook, K. [1981], "On Non-Preemptive Strategies in Stochastic Scheduling," **Naval Research Logistics Quarterly**, Vol. 28, pp. 289–300.

Hopp, W., J. Bean, and R. Smith [1984], "Non-Homogeneous Markov Decision Processes," Technical Report 84–24, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan 48109.

McKenzie, L. [1976], "Turnpike Theory," **Econometrica**, Vol. 44, pp. 841–864.

Pinedo, M. [1983], "Stochastic Scheduling with Release Dates and Due Dates," **Operations Research**, Vol. 31, pp. 559–572.

Pinedo, M. and S. Ross [1980], "Scheduling Jobs Subject to Non-Homogeneous Poisson Shocks," **Management Science**, Vol. 26, pp. 1250-1257.

Ramsey, F. [1928],"A Mathematical Theory of Savings," **Economic Journal**, Vol. 38, pp. 543–559.

FIGURE 1: Optimal Paths

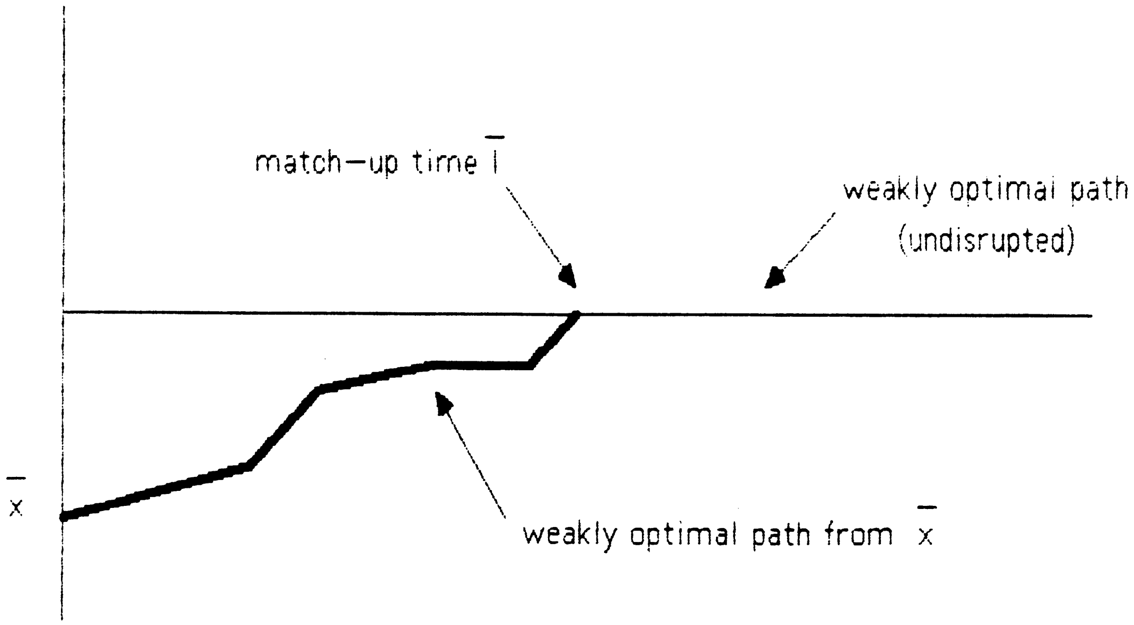
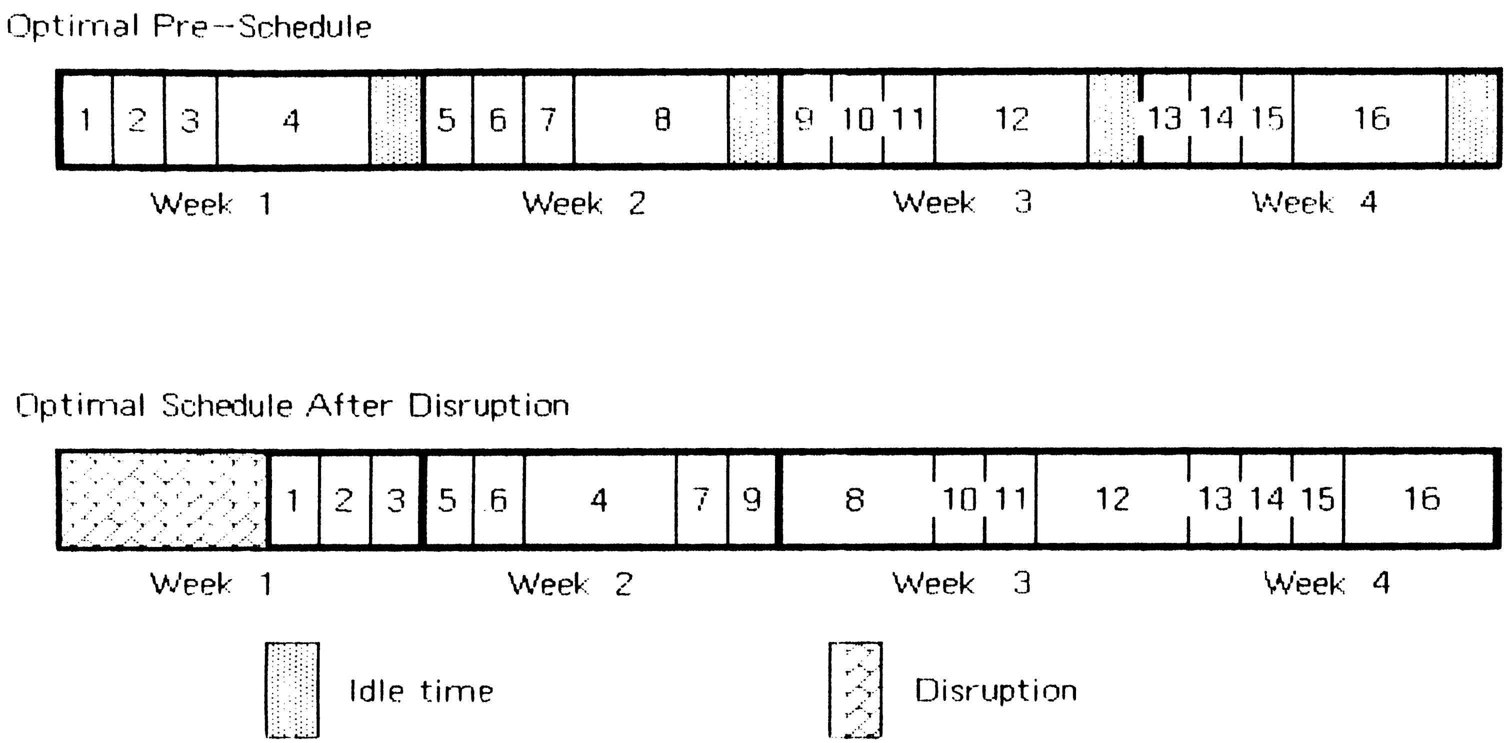


FIGURE 2: Match-Up Example



Optimal Pre-Schedule

Optimal Schedule After Disruption