PROBABILISTIC ANALYSIS AND PRACTICAL ALGORITHMS FOR THE FLOW SHOP WEIGHTED COMPLETION TIME PROBLEM

PHILIP KAMINSKY

University of California, Berkeley, California

DAVID SIMCHI-LEVI

Northwestern University, Chicago, Illinois

(Received February 1996; revisions received October 1996, April 1997; accepted April 1997)

In the flow shop weighted completion time problem, a set of jobs has to be processed on m machines. Every machine has to process each one of the jobs, and every job has the same routing through the machines. The objective is to determine a sequence of the jobs on the machines so as to minimize the sum of the weighted completion times of all jobs on the final machine. In this paper, we present a characterization of the asymptotic optimal solution value for general distributions of the job processing times and weights. In particular, we show that the optimal objective value of this problem is asymptotically equivalent to certain single and parallel machine scheduling problems. This characterization leads to a better understanding of the effectiveness of the celebrated weighted shortest processing time algorithm, as well as to the development of an effective algorithm closely related to the profile fitting heuristic, which was previously utilized for flow shop makespan problems. Computational results show the effectiveness of WSPT and this modified profile fitting heuristic on a set of random test problems.

In the m-**machine flow shop problem**, a set of jobs, each consisting of m operations, must be sequentially processed on m machines. Each machine can handle at most one job at a time, and a job can only be processed on one machine at a time. The jobs have to be processed on each of the machines without preemption, and every machine serves the arriving jobs in a first come first served fashion. Given the processing times of each of the jobs on each of the machines, and weights associated with each of the jobs, the **Flow Shop Weighted Completion Time Problem** involves determining a sequence of the jobs on the machines so as to minimize the average, or equivalently the sum, of the weighted completion times of the jobs on the final machine in the sequence. It is well known (see Garey et al. 1976) that this problem is NP-hard even in the two-machine case with all weights equal.

The majority of flow-shop related research has focused on minimizing the makespan, that is, minimizing the time it takes to complete processing all jobs. This is due to the fact that individual job-related objectives, such as mean completion time, are very difficult to analyze, and in fact, as Pinedo (1995) points out, "makespan results are already relatively hard to obtain." Nevertheless, individual job related objectives capture important real-life managerial scheduling concerns that are not reflected in the makespan and similar objectives (see, for example, Morton and Pentico 1993).

Previous research on the Flow Shop Mean Completion Time Problem has typically focused on branch-and-bound or local search strategies, sometimes with as many as 10 machines and 50 jobs, but most often with only 2 machines. For instance, Ignall and Schrage (1965) first applied branch and bound to small size flow shop problems, while Krone and Steiglitz (1974) applied local search techniques. Kohler and Steiglitz (1975) combined these approaches to solve two-machine problems of up to 15 jobs to optimality, and of up to 50 jobs approximately. Szwarc (1982) and Adiri and Amit (1984) identified various properties of this problem as well as classes of more easily solvable special cases. Van de Velde (1990) utilized Langrangean relaxation to determine lower bounds when building the branch-and-bound tree, and effectively solved problems with 2 machines and up to 20 jobs to optimality. Finally, Bhaskaran and Pinedo (1992) and Morton and Pentico (1993) suggest a variety of *dispatch rules* as a way to solve real-world industrial flow shop problems.

In this paper, we take a different approach to the flow shop weighted completion time problem. Utilizing probabilistic analysis techniques similar to those that have recently proved effective for large scale vehicle routing problems (see Bramel and Simchi-Levi 1995 and Bramel and Simchi-Levi 1996), we characterize the underlying structure of the asymptotic optimal solution to the flow shop weighted completion time problem. Interestingly, we demonstrate that the asymptotic optimal objective value of this problem is directly related to the asymptotic objective value of *certain single and parallel machine scheduling problems*. By-products of the analysis are a better understanding of the effectiveness of the celebrated **Weighted Shortest Processing Time** (WSPT) first rule, as well as the development of an algorithm based on the **Profile Fitting Heuristic** proposed by McCormick et al. (1989) for the Flow Shop Makespan Problem with Blocking.

Subject classifications: Production/scheduling, multiple machine sequencing: flow shop weighted completion time problem. Analysis of algorithm: probabilistic analysis.
Area of review: OPTIMIZATION.

0030-364X/98/4606-0872 $05.00

It is worth pointing out that much of this research was motivated by the success we had (see Kaminsky and Simchi-Levi 1998), using the WSPT algorithm to solve some large-scale, industrial scheduling problems. In fact, this dispatch rule proved to be much more effective than many more complex algorithms that we tested for this industrial problem. Indeed, computational results presented in this paper with random test problems show that WSPT performs well relative to a lower bound for the Flow Shop Weighted Completion Time Problem. We note that in the context of the flow shop model analyzed in this paper, the WSPT first rule sequences the jobs in decreasing order of the ratio of the job weight to the job *total* processing time.

To put our work and results in perspective it is important to describe some related work on probabilistic analysis of algorithms for machine scheduling problems. As far as we are aware, most of this work has focused on the **Parallel Machine Scheduling Problem**, in which each job has to be processed on one out of m identical machines and the objective is to minimize the makespan. For instance, Coffman et al. (1982), Loulou (1984), and Frenk and Rinnooy Kan (1987) have analyzed the performance of the **Longest Processing Time** first rule. Spaccamela et al. (1992) have analyzed the same model when the objective is to minimize the *weighted completion time* and demonstrate that in this case the **Weighted Shortest Processing Time** first rule is asymptotically optimal. Webster (1993) extends these results to some instances of parallel machines with different speeds. Finally, Chan et al. (1996) use probabilistic analysis to characterize the effectiveness of linear programming relaxations of set partitioning formulations of this model. A departure from this line of problems is presented in Ramudhin et al. (1996), in which the two-machine flow shop model is analyzed when the objective is to minimize the makespan. They characterize the expected behavior of a variety of strategies including optimal and approximate algorithms.

In the next section we provide a detailed description of the model analyzed together with our main result.

1. THE MODEL AND THE MAIN RESULT

To formally present the model, consider a set of n jobs that have to be processed on m machines. Job i, $i = 1, 2, \ldots, n$, has a processing time t_i^l on machine l, $l = 1, 2, \ldots, m$, and an associated weight w_i. The processing times are independent and identically distributed random variables, defined on the interval (0, 1]. Similarly, the weights are independent and identically distributed random variables, defined on the interval (0, 1].

Each job must be processed without preemption on each of the machines sequentially. That is, each job must be processed on machine 1 through machine m in that order. Jobs are available for processing at time zero, and with the exception of the first machine, all other machines process the jobs in a first-come-first-served manner, a so-called **permutation** schedule. Also, there is unlimited intermediate storage between successive machines. The objective is to determine a *schedule*, or sequence of jobs, such that the total weighted completion times of all the jobs on the final machine is minimized. We call this problem Problem P and use Z^* to denote its optimal objective function value. That is, Z^* is the minimum possible total weighted completion time of all jobs in Problem P. Similarly, given a heuristic H for the Flow Shop Weighted Completion Time Problem, we use Z^H to denote the sum of the weighted completion time in the resulting schedule.

Associated with an instance of the Flow Shop Weighted Completion Time Problem is the following parallel machine scheduling model. Given job i, $i = 1, 2, \ldots, n$, with processing times $t_i^1, t_i^2, \ldots, t_i^m$ on machine $1, 2, \ldots, m$, respectively, let $t_i = \sum_{l=1}^{m} t_i^l$. Consider a parallel machine scheduling problem with k machines and n tasks each having a processing time t_i and a weight w_i, $i = 1, 2, \ldots, n$. The objective in the parallel machine scheduling problem is to assign each task to a single machine so as to minimize the sum of the weighted completion times of all tasks. We refer to this parallel machine scheduling problem as Problem P_k with Z_k^* as its optimal solution value, the minimum total weighted completion time of all the tasks in the parallel machine scheduling problem with k machines. Thus, Z_m^* is the optimal solution to Problem P_m, the parallel machines scheduling problem with m machines and n tasks each having a processing time t_i and weight w_i, $i = 1, 2, \ldots, n$. Similarly, Z_1^* is the optimal solution to Problem P_1, the single machine scheduling problem with n tasks each having a processing time t_i and weight w_i, $i = 1, 2, \ldots, n$. Unlike the Flow Shop Weighted Completion Time Problem and the associated parallel machine scheduling problem, the optimal solution to the single machine scheduling problem is easily obtained via the WSPT first rule; see, for example, Pinedo (1995).

Recently, Spaccamela et al. (1992) established the equivalence between the single and parallel machine problems. Their result, translated to our model, is stated in the following theorem.

Theorem 1.1. *Let the processing times* $t_i^1, t_i^2, \ldots, t_i^m$, $i = 1, 2, \ldots, n$, *be independent and identically distributed random variables defined on* (0, 1]. *Let the weights* w_i, $i = 1, 2, \ldots, n$, *be independent and identically distributed random variables defined on* (0, 1]. *Then with probability one we have*

$$\lim_{n\to\infty} \frac{Z_m^*}{n^2} = \lim_{n\to\infty} \frac{Z_1^*}{mn^2} = \theta,$$

for some constant θ.

In fact, the Spaccamela et al. result is more general; it allows for unbounded random variables provided that certain restrictions are met. In addition, they also characterize the constant θ and express it as an expected value of a stylized kernel function. Kaminsky (1997) provides a closed form expression for the special case when all the weights, w_i, are equal.

Building on their result, we prove Theorem 1.2.

Theorem 1.2. *Let the processing times* $t_i^1, t_i^2, \ldots, t_i^m$, $i = 1, 2, \ldots, n$, *be independent random variables having the same continuous distribution with bounded density* $\phi(\cdot)$ *defined on* $(0, 1]$. *Let the weights* w_i, $i = 1, 2, \ldots, n$, *be independently and identically distributed according to a cumulative distribution function* $\Phi(\cdot)$ *defined on* $(0, 1]$. *Then with probability one we have*

$$\lim_{n\to\infty} \frac{Z^*}{n^2} = \lim_{n\to\infty} \frac{Z^*_m}{n^2} = \lim_{n\to\infty} \frac{Z^*_1}{mn^2} = \theta,$$

for some constant θ.

Theorem 1.2 thus implies that asymptotically there is no difference between the optimal solution to Problem P, the Flow Shop Weighted Completion Time Problem, and the optimal solution to its associated parallel machine scheduling problem, Problem P_m. Such an insight is useful in the context of capital investment issues in which a decision is being made between machines that perform sequential operations and multipurpose machines that operate in parallel.

It is also interesting to note that Theorem 1.2 characterizes the asymptotic equivalence of three models, one of which (the single machine model) can easily be solved in polynomial time, one of which can be either easy to solve or NP-hard, depending on whether or not the weights are equal (the parallel machine model), and one of which is NP-hard even in the case of equal weights and two machines (the flow shop model).

To prove Theorem 1.2, we start in Section 3 by presenting a specialized model that captures the essential ideas of our proof. In Section 4 we build on this analysis, providing a formal proof for Theorem 1.2. In Section 5 we demonstrate that the results and the accompanying analysis lead to an understanding of the effectiveness of the Weighted Shortest Processing Time Rule for the Flow Shop Weighted Completion Time Problem. Computational evidence with randomly generated instances shows that in many cases WSPT is very effective. Finally, the structural knowledge gained in this analysis indicates that a modified version of the Profile Fitting Heuristic developed by McCormick et al. (1989) for the Flow Shop Makespan problem will also be effective for the Weighted Completion Time Problem, and this is supported by some computational testing of this heuristic.

2. PRELIMINARIES

In this section we develop a fundamental, but simple, lower bound on the optimal solution to Problem P, Z^*, which we use throughout the paper. This lower bound is directly related to the optimal solution to Problem P_1, Z_1^*. In later sections we show that this lower bound is asymptotically tight.

Lemma 2.1. *Consider Problem* P, *the general Flow Shop Weighted Completion Time Problem, and its associated single machine scheduling problem, Problem* P_1. *For every instance we have,*

$$\frac{1}{m} Z^*_1 \leqslant Z^*.$$

Proof. Given the optimal sequence to Problem P, index the jobs according to their departure time from the last machine, starting with the index [1] and finishing with the index $[n]$. Note that this is not necessarily a WSPT sequence. Let $C_{[i]}$ be the completion time of the ith job that departs from the last machine in that sequence. Let $t_{[i]}$ be the total processing times of job $[i]$ on all the m machines. These definitions imply that

$$mC_{[i]} \geqslant \sum_{j=1}^{i} t_{[j]},$$

since $mC_{[i]}$ is the total time available on all the machines up to time $C_{[i]}$ while $\sum_{j=1}^{i} t_{[j]}$ is the time used by jobs [1], [2], $\ldots$, $[i]$. Rearranging this inequality and multiplying by the weight of job $[i]$ gives

$$w_{[i]}C_{[i]} \geqslant w_{[i]} \frac{1}{m} \sum_{j=1}^{i} t_{[j]}.$$

Summing over all of the jobs we see that

$$\sum_{i=1}^{n} w_{[i]}C_{[i]} \geqslant \frac{1}{m} \sum_{i=1}^{n} w_{[i]} \sum_{j=1}^{i} t_{[j]},$$

and since the WSPT gives the optimal solution to the single machine problem, we have

$$Z^* = \sum_{i=1}^{n} w_{[i]}C_{[i]} \geqslant \frac{1}{m} \sum_{i=1}^{n} w_{[i]} \sum_{j=1}^{i} t_{[j]} \geqslant \frac{1}{m} Z^*_1,$$

which completes the proof. □

3. THE CYCLIC DISCRETE MODEL

Our strategy in proving Theorem 1.2 is to introduce a specific discrete model, called the *Cyclic Discrete* model, with a finite number of different possible processing times, and with a special relationship between certain subsets of the jobs. For this specialized model, we prove a result analogous to Theorem 1.2, by utilizing the characteristics of its special structure. Then, in Section 4, we use this result in the analysis of Problem P by showing that the optimal solution to the flow shop problem can be bounded from above by the optimal solution of an associated *Cyclic Discrete* model. This, together with the lower bound developed in Lemma 2.1, will prove our main result.

It is important to point out that the *Cyclic Discrete* model is not only essential to the proof of Theorem 1.2, but, as discussed in Section 5, it also provides insight into the *structure* of the algorithms needed to solve large-scale machine scheduling problems. Indeed, this insight is used in our development of a new algorithm for the general flow shop weighted completion time problem. This *Cyclic Discrete* model is defined below.

Consider an m machine flow shop model for which the objective is to minimize the sum of the weighted completion times. Each job has an associated vector $(t^1, t^2, \ldots, t^m)$ where t^i, $t^i \in (0, 1]$, is the processing time on the ith machine. We call this vector a *time assignment vector*. The *total processing time* of a job with time assignment vector $(t^1, t^2, \ldots, t^m)$ is the quantity $\sum_{l=1}^m t^l$. Each job also has an associated weight, defined on (0, 1].

We say that two jobs are *identical* if their associated weights and *time assignment vectors* are equal, element wise, and we call a set of identical jobs, which can all be represented by the same *time assignment vector* and weight, a *job type*.

Given a *job type*, we can construct a number of new job types through a cyclic shift of the elements in its *time assignment vector*. That is, given a *job type* with time assignment vector $(t^1, t^2, \ldots, t^m)$ and weight w, new *shifted job types* are created by shifting the processing times over one machine in a cyclic manner, and using the same weight w. In that process we create *job types* with weight w and the following time assignment vectors:

$$(t^2, t^3, \ldots, t^m, t^1), (t^3, t^4, \ldots, t^m, t^1, t^2), \ldots, (t^m, t^1, t^2, \ldots, t^{m-1}).$$

Of course, if some of the processing times t^l, $l = 1, 2, \ldots, m$, are equal, some of the *job types* created in the process may be identical. If, on the other hand, the processing times t^l, $l = 1, 2, \ldots, m$, are all different, the shifted cyclic process will generate $m - 1$ new, nonidentical, *job types*.

We define *group type* g_j to consist of a *job type*, which we call j_1, and its $m - 1$ *cyclic shifted job types,* $j_2, j_3, \ldots, j_m$, where *job type* j_2 is shifted left one position from j_1, j_3 is shifted two positions from j_1, and so on, and all of the *job types* have the same associated weight, w_j. Let t^i_{jk} represent the processing time on the ith machine of the kth *job type* in group g_j, for $i = 1, 2, \ldots, m$, $k = 1, 2, \ldots, m$. Given j_1, *shifted job type* j_2 has an associated time assignment vector whose elements are

$$t^1_{j2} = t^2_{j1},\ t^2_{j2} = t^3_{j1},\ \ldots,\ t^{m-1}_{j2} = t^m_{j1},\ t^m_{j2} = t^1_{j1}.$$

Similarly, *shifted job type* j_3 is the vector with elements

$$t^1_{j3} = t^3_{j1},\ t^2_{j3} = t^4_{j1},\ \ldots,\ t^{m-2}_{j3} = t^m_{j1},\ t^{m-1}_{j3} = t^1_{j1},\ t^m_{j3} = t^2_{j1}.$$

The remaining *job types* in the group g_j are created in the same manner. Thus, each *group type* g_j consists of m job types, each of which has the same *total* processing time, t_j. To simplify the analysis which follows, we restrict ourselves in this section to *job types* with associated *time assignment vectors* such that *no two elements of the vector are equal*, and no two *job types* are identical.

Now, consider a model in which there is a *finite* number, s, of *group types*. Let n_j be the number of jobs of each of the *job types* within group g_j, for $j = 1, 2, \ldots, s$. Thus, each of *job types* in group g_j, j_i, $i = 1, 2, \ldots, m$ has the same number of jobs assigned to it, so $n = m \sum_{j=1}^s n_j$ is the total number of jobs, out of which mn_j are associated with *group type* g_j. Let Z^* be the optimal solution to this m machine flow shop problem, where the objective is to minimize the sum of the weighted completion time. In what follows we refer to this problem as the *original Cyclic Discrete* problem.

Define now a corresponding parallel machine scheduling problem in exactly the same way it is done in Section 1, by associating a task with each one of the jobs of the original problem.

Given the n tasks associated with the n jobs of the original cyclic discrete model, recall that Z^*_m is the optimal solution to the parallel machine scheduling problem with m machines and n tasks, while Z^*_1 is the optimal solution to a single machine scheduling problem with n tasks. The optimal solution to the single machine scheduling problem is easily obtained by using the WSPT first rule. Consequently, we order the groups in a nonincreasing order of the ratio of their weights to their processing times

$$\frac{w_1}{t_1} \geqslant \frac{w_2}{t_2} \geqslant \cdots \geqslant \frac{w_s}{t_s}.$$

In the probabilistic analysis that follows, we consider a *Cyclic Discrete* model in which groups of m jobs are added to the model by selecting a *group type* g_j with probability p_j, for $j = 1, 2, \ldots, s$, and then generating m jobs, one for each *job type* within that group. That is, with probability one, we have $p_j = \lim_{n\to\infty} n_j/\sum_{l=1}^s n_l$ for $j = 1, 2, \ldots, s$.

We prove the following theorem.

Theorem 3.1. *The optimal solutions to the cyclic discrete model and its corresponding parallel machine scheduling problem satisfy with probability one*

$$\lim_{n\to\infty} \frac{Z^*}{n^2} = \lim_{n\to\infty} \frac{Z^*_m}{n^2} = \lim_{n\to\infty} \frac{Z^*_1}{mn^2} = \theta',$$

where

$$\theta' = \frac{1}{2m}\left[\sum_{i=1}^{s}\sum_{j=1}^{i-1} w_i p_i p_j t_j + \sum_{i=1}^{s}\sum_{j=i+1}^{s} w_j p_i p_j t_i + \sum_{i=1}^{s} w_i p_i^2 t_i\right].$$

To prove Theorem 3.1, we construct upper and lower bounds on Z^* that converge to the same value. Surprisingly, this value is precisely the asymptotic optimal solution of the corresponding parallel machine scheduling problem.

3.1. Upper Bound

Order the group types in a nonincreasing order of the quantities w_j/t_j, the ratio of their weights to their total processing times and let this ordering be $g_1, g_2, \ldots, g_s$. Consider the following strategy for the original *Cyclic Discrete m* machine scheduling problem. Starting with the first group type, g_1, its corresponding jobs types, $1_1, 1_2, \ldots, 1_m$,

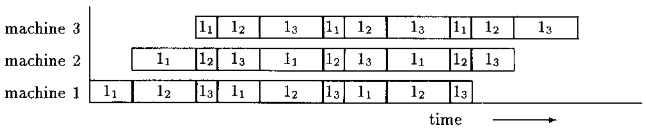

Figure 1. The first group.

and the associated mn_1 jobs, schedule all these jobs by cycling through the job types in the order

$$1_1, 1_2, 1_3, \ldots, 1_m,$$

each time assigning a single job from a different job type, until all jobs are assigned. A sample sequence with three machines and $n_1 = 3$ is illustrated in Figure 1.

After all the jobs of group type g_1 are completely scheduled, schedule all the mn_2 jobs of group type g_2, starting at the time the last job from the previous job type ended on the *last* machine. This implies that there will be idle time on each machine between the time the last job of group type g_1 departed that machine and the time the first job of group type g_2 started on it. Figure 2 illustrates the previous example after the addition of the second group.

Continue scheduling the remaining jobs in this way until all jobs are scheduled. We refer to this strategy as the *m-machine Interchange Strategy* and denote its objective value by Z^{INT}. Given group type g_j, $j = 1, 2, \ldots, s$, and its associated m job types, let

$$L_j = n_j t_j + t_j^1 + t_j^2 + \cdots + t_j^{m-1}.$$

Thus, L_j is the total time it would take to complete processing all jobs of group type g_j using the interchange strategy, if there were only those jobs. Similarly, given group type g_j, $j = 1, 2, \ldots, s$, its associated job types and all their corresponding jobs, let

$$F_j = t_j \frac{(n_j + 1)n_j}{2}. \tag{1}$$

Thus, F_j is the sum of the completion times of all jobs of type j_1 if the jobs in g_j are scheduled using the above interchange strategy, and when no other group types exist.

We now proceed to find an upper bound on Z^{INT}. For this purpose, observe that the ith job of type 1_1 departs the last machine at time it_1, for $i = 1, 2, \ldots, n_1$. Similarly, the ith job of type 1_u, $u = 2, 3, \ldots, m$, departs the last machine no later than $it_1 + t_1$. Thus, the weighted sum of the completion times of all jobs of group type g_1 is no more than

$$w_1[mF_1 + (m-1)t_1 n_1].$$

The jobs of group type g_2 are scheduled after all the jobs of group type g_1 are completed. Thus, L_1, the time the last job of group type g_1 completed processing on machine m, is the time the first job of group type g_2 starts on the first machine. Consequently, the weighted sum of completion time of the jobs of group type g_2 is no more than

$$w_2[mn_2L_1 + mF_2 + (m-1)t_2n_2].$$

Following a similar pattern for the remaining group types, we determine an upper bound on the sum of completion times for the interchange strategy for all of the n jobs.

$$\begin{aligned} Z^* \leqslant\ & w_1(mF_1 + (m-1)t_1n_1) \\ & + w_2mn_2L_1 + w_2(mF_2 + (\mathrm{m}-1)t_2n_2) \\ & + w_3mn_3(L_1 + L_2) + w_3(mF_3 + (m-1)t_3n_3) + \\ & \vdots \\ & + w_smn_s(L_1 + L_2 + \cdots + L_{s-1}) \\ & + w_s(mF_s + (m-1)t_sn_s). \end{aligned} \tag{2}$$

Finally, dividing the inequality by n^2, taking the number of jobs, n, to infinity, noting that with probability one

$$p_j = \lim_{n\to\infty} mn_j/n, \quad \forall j = 1, 2, \ldots, s,$$

and using the definition of L_j, $j = 1, 2, \ldots, s$, we get with probability one:

$$\begin{aligned} \overline{\lim_{n\to\infty}} \frac{2mZ^*}{n^2} & \leqslant w_1t_1p_1^2 + 2w_2p_2p_1t_1 + w_2t_2p_2^2 \\ & + 2w_3p_3(p_2t_2 + p_1t_1) + w_3t_3p_3^2 + \cdots \\ & + 2w_sp_s(p_1t_1 + p_2t_2 + \cdots + p_{s-1}t_{s-1}) \\ & + w_st_sp_s^2 \\ & = 2m\theta'. \end{aligned} \tag{3}$$

3.2. Lower Bound

We use the lower bound developed in Section 2. Consider the original cyclic discrete problem and its associated single machine scheduling problem. In the latter model we have mn_j jobs each having a processing time t_j, and weight w_j, for $j = 1, 2, \ldots, s$. Minimizing the total weighted completion time of n jobs on a single machine is obtained using the WSPT first rule. Let

$$G_j = t_j \frac{(mn_j + 1)mn_j}{2}, \quad \forall j = 1, 2, \ldots, s.$$

The optimal objective value of the single machine problem is clearly

$$Z_1^* = \sum_{j=1}^{s} w_jG_j + m^2 \sum_{k=2}^{s} w_kn_k \left(\sum_{i=1}^{k-1} n_it_i \right).$$

Dividing by mn^2 and taking the limit we get with probability one

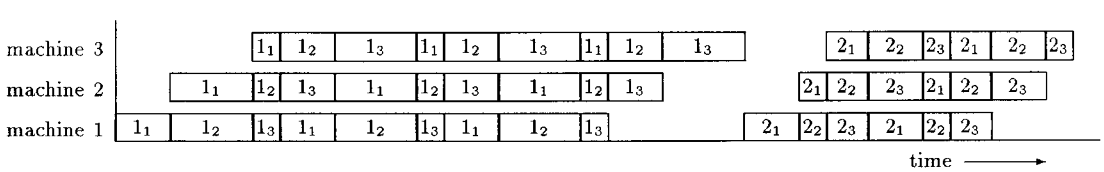

Figure 2. The first two groups.

$$\lim_{n\to\infty} \frac{Z_1^*}{mn^2}$$
$$= \frac{1}{2m} [w_1 t_1 p_1^2 + 2w_2 p_2 p_1 t_1 + w_2 t_2 p_2^2 + 2w_3 p_3 (p_2 t_2 + p_1 t_1) + w_3 t_3 p_3^2 + \cdots + 2w_s p_s (p_1 t_1 + p_2 t_2 + \cdots + p_{s-1} t_{s-1}) + w_s t_s p_s^2] = \theta'. \tag{4}$$

Combining this result with Lemma 2.1 and inequality (3) proves that almost surely the *Interchange Strategy* is asymptotically optimal, and that the asymptotic optimal objective value satisfies, with probability one:

$$\lim_{n\to\infty} \frac{Z^*}{n^2} = \lim_{n\to\infty} \frac{Z_1^*}{mn^2} = \theta'. \tag{5}$$

This, together with the result by Spaccamela et al. (1992) listed in Theorem 1.1, completes the proof of Theorem 3.1.

Observe that the *Interchange Strategy* orders the jobs according to the WSPT first rule. On the other hand, not every sequence generated by the WSPT first rule follows the interchange strategy. We thus conclude the following corollary.

Corollary 3.2. *There exists a WSPT sequence that is asymptotically optimal for the Cyclic Discrete Flow Shop Weighted Completion Time Problem.*

4. PROOF OF THE MAIN THEOREM

We prove Theorem 1.2 by constructing a number of closely related discretized versions of Problem P. These discretized models allow us to develop upper bounds on Z^*/n^2 that are related to the *Cyclic Discrete Model* analyzed in the previous section, and that converge to the lower bound developed in Lemma 2.1.

To discretize the problem, we subdivide the (0, 1] interval into s subintervals, each of length ϵ. We use A_l, $l = 1, 2, \ldots, s$, to denote the lth subinterval, that is, $A_l = ((l-1)\epsilon, l\epsilon]$. For every job i in Problem P, $i = 1, 2, \ldots, n$, and machine k, $k = 1, 2, \ldots, m$, such that $t_i^k \in A_l$ for some l, $l = 1, 2, \ldots, s$, and $w_i \in A_j$ for some j, $j = 1, 2, \ldots, s$, we round its processing time, t_i^k, up to the value $l\epsilon$ and its weight, w_i, up to the value $j\epsilon$. The resulting problem is an m machine flow shop problem for which the objective is to minimize the total weighted completion time of all the n jobs. We refer to this problem as Problem $\bar{P}_D$ whose optimal objective function value is $\bar{Z}_D^*$. It is easy to see that

$$Z^* \leq \bar{Z}_D^*. \tag{6}$$

Since in Problem $\bar{P}_D$ processing times can take only discrete values, we can construct an associated cyclic discrete problem called Problem $\bar{P}_{CD}$ whose optimal solution is $\bar{Z}_{CD}^*$. As in the previous section, let a *job type* be represented by an associated weight and *time assignment vector* $(t^1, t^2, \ldots, t^m)$. In Problem $\bar{P}_{CD}$, for every k, $k = 1, 2, \ldots, m$, we have $t^k = l\epsilon$ for some l, $l = 1, 2, \ldots, s$, and we consider **only** *time assignment vectors* **such that each vector has no two equal elements**. We partition the set of all *job types* with the above property into groups $g_1, g_2, \ldots, g_G$ such that each group includes **all** the *job types* that are obtained by a cyclic shift of all of the *job types* in the group, and all *job types* within a group have the same weight. Clearly each such group consists of exactly m *job types*, and all the *job types* within a single group correspond to the *job types* as defined in Section 3. Let $n_{g_i}^l$ be the number of jobs in Problem $\bar{P}_D$ whose processing times and weight is represented by the lth *job type* of group g_i and its associated weight, $l = 1, \ldots m$, and $i = 1, 2, \ldots, G$. Let

$$\tilde{n} = n - \sum_{j=1}^{G} \sum_{l=1}^{m} n_{g_i}^l,$$

that is, $\tilde{n}$ is the number of jobs in Problem $\bar{P}_D$, each of which has at least two machines on which its processing times are equal.

In the new problem, Problem $\bar{P}_{CD}$, we assign exactly

$$n_{g_i} = \min_{l=1,\ldots,m} \{n_{g_i}^l\},$$

jobs to each one of the job types associated with group g_i. Let $\bar{Z}_{CD}^*$ be the optimal solution value of the resulting problem and observe that this problem is a *Cyclic Discrete* problem as defined in Section 3.

Our objective is to use Problem $\bar{P}_{CD}$ in two ways: to construct an upper bound on $\bar{Z}_D^*$, and to relate this upper bound to Z_1^*, the optimal solution to Problem P_1.

We use the optimal solution to Problem $\bar{P}_{CD}$ to construct an upper bound on the optimal solution of Problem $\bar{P}_D$, as follows. Start by scheduling jobs according to the optimal solution to Problem $\bar{P}_{CD}$, and then schedule all the remaining

$$\tilde{n} + \sum_{i=1}^{G} \sum_{l=1}^{m} (n_{g_i}^l - n_{g_i})$$

jobs at the end of the sequence. Hence,

$$\bar{Z}_D^* \leq \bar{Z}_{CD}^* + \left(\sum_{i=1}^{n} \sum_{k=1}^{m} t_i^k \right) \left[\tilde{n} + \sum_{i=1}^{G} \sum_{l=1}^{m} (n_{g_i}^l - n_{g_i}) \right], \tag{7}$$

since the weight of each job is bounded by one.

In order to relate Problem $\bar{P}_{CD}$ to Problem P_1, we begin by considering an instance of Problem $\bar{P}_{CD}$ and constructing an instance of the related single machine total completion time problem in the same way that Problem P_1 is constructed from Problem P, in Section 1. That is, every job i in Problem $\bar{P}_{CD}$ with processing time t_i^l on machine l, $l = 1, 2, \ldots, m$, has a corresponding task in the new instance with processing time $\sum_{l=1}^{m} t_i^l$. We refer to this single machine total completion time problem as Problem $\bar{P}_{1CD}$, and use $\bar{Z}_{1CD}^*$ to denote its optimal solution, the minimum total completion time among all possible schedule for that problem.

Theorem 3.1 tells us that the optimal solutions of Problems $\bar{P}_{CD}$ and $\bar{P}_{1CD}$ are closely related. That is, if n_{cd} is the number of jobs in Problem $\bar{P}_{CD}$, then with probability one we have

$$\lim_{n_{CD}\to\infty} \frac{\bar{Z}^*_{CD}}{n^2_{CD}} = \lim_{n_{CD}\to\infty} \frac{\bar{Z}^*_{1CD}}{mn^2_{CD}}. \tag{8}$$

Dividing Equation (7) by n^2, taking the limit as n goes to infinity, and using Equations (6) and (8), we obtain

$$\begin{aligned}\overline{\lim_{n\to\infty}} \frac{Z^*}{n^2} &\leq \overline{\lim_{n\to\infty}} \frac{\bar{Z}^*_{CD}}{n^2} + \overline{\lim_{n\to\infty}} \frac{1}{n^2}\left(\sum_{i=1}^{n}\sum_{k=1}^{m} t_i^k\right) \\ &\quad \cdot \left[\tilde{n} + \sum_{i=1}^{G}\sum_{l=1}^{m} (n^l_{g_i} - n_{g_i})\right] \\ &\leq \overline{\lim_{n\to\infty}} \frac{\bar{Z}^*_{1CD}}{mn^2} + \overline{\lim_{n\to\infty}} \frac{1}{n^2}\left(\sum_{i=1}^{n}\sum_{k=1}^{m} t_i^k\right) \\ &\quad \cdot \left[\tilde{n} + \sum_{i=1}^{G}\sum_{l=1}^{m} (n^l_{g_i} - n_{g_i})\right].\end{aligned} \tag{9}$$

In order to obtain the desired asymptotic results, we need to show that the second term on right hand side of the above upper bound is almost surely $O(\epsilon)$. For this purpose, note that the number of groups, G, in Problem $\bar{P}_{CD}$ is only a function of s, the number of subintervals, and m, the number of machines, but not a function of n, the number of jobs, and hence with probability one,

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{G}\sum_{l=1}^{m} (n^l_{g_i} - n_{g_i}) = 0. \tag{10}$$

In addition, recall that $\tilde{n}$ is the number of jobs in Problem $\bar{P}_D$ such that each job has a corresponding *time assignment vector* for which at least two machines have equal processing times. How many such *time assignment vectors* exist? It is easy to see that this number is no more than

$$\binom{m}{2} s^{m-1}.$$

Since the distribution $\phi(\cdot)$ is bounded, there exists a constant K such that $\phi(u) \leq K$ for every $u \in (0, 1]$. Hence, the probability that a job in Problem $\bar{P}_D$ has a corresponding *time assignment vector* for which at least two machines have equal processing times is no more than

$$\binom{m}{2} s^{m-1}(K\epsilon)^m.$$

This, together with $s\epsilon = 1$, implies that with probability one,

$$\lim_{n\to\infty} \frac{\tilde{n}}{n} = \binom{m}{2} s^{m-1}(K\epsilon)^m = O(\epsilon). \tag{11}$$

Finally, it is easy to see that almost surely

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{m} t_i^k = mE[u],$$

where $E[u]$ is the expected value of the random variable u whose density function is $\phi(\cdot)$. Consequently, with probability one we have

$$\lim_{n\to\infty} \frac{1}{n}\left(\sum_{i=1}^{n}\sum_{k=1}^{m} t_i^k\right)\left[\tilde{n} + \sum_{i=1}^{G}\sum_{l=1}^{m} (n^l_{g_i} - n_{g_i})\right] = O(\epsilon), \tag{12}$$

and therefore, using Equation (9), we get that almost surely

$$\overline{\lim_{n\to\infty}} \frac{Z^*}{n^2} \leq \overline{\lim_{n\to\infty}} \frac{\bar{Z}^*_{1CD}}{mn^2} + O(\epsilon). \tag{13}$$

To complete the proof of Theorem 1.2, we relate $\bar{Z}^*_{1CD}$ to Z^*_1, the optimal solution to the single machine problem, Problem P_1, defined in Section 1. For this purpose, observe that every task in Problem $\bar{P}_{1CD}$ has a corresponding task in Problem P_1. In addition, the processing time of every task in Problem $\bar{P}_{1CD}$ is no more than $m\epsilon$ larger than the processing time of the corresponding task in Problem P_1. Similarly, the weight of every task in Problem $\bar{P}_{1CD}$ is no more than ϵ larger than the weight of the corresponding task in Problem P_1. Consequently,

$$\bar{Z}^*_{1CD} \leq Z^*_1 + \frac{(n+1)n}{2}(m+1)\epsilon,$$

and therefore,

$$\overline{\lim_{n\to\infty}} \frac{Z^*}{n^2} \leq \overline{\lim_{n\to\infty}} \frac{\bar{Z}^*_{1CD}}{mn^2} + O(\epsilon) \leq \overline{\lim_{n\to\infty}} \frac{Z^*_1}{mn^2} + O(\epsilon). \tag{14}$$

On the other hand, Lemma 2.1 tells us that with probability one we have

$$\underline{\lim_{n\to\infty}} \frac{Z^*}{n^2} \geq \lim_{n\to\infty} \frac{Z^*_1}{mn^2}. \tag{15}$$

Combining Equations (14) and (15), and choosing ϵ small enough, shows that with probability one we have

$$\lim_{n\to\infty} \frac{Z^*}{n^2} = \lim_{n\to\infty} \frac{Z^*_1}{mn^2}.$$

Finally, using the result of Spaccamela et al. (1992) as stated in Theorem 1.1, we relate Z^*_1 to Z^*_m and show that there exists a constant θ such that with probability one

$$\lim_{n\to\infty} \frac{Z^*}{n^2} = \lim_{n\to\infty} \frac{Z^*_m}{n^2} = \lim_{n\to\infty} \frac{Z^*_1}{mn^2} = \theta.$$

5. ALGORITHMS AND COMPUTATIONAL RESULTS

5.1. Weighted Shortest Processing Time Rule

The analysis in the previous sections indicates that the WSPT first rule has the potential to be quite effective for the Flow Shop Weighted Completion Time Problem; Corollary 3.2 tells us that there is a WSPT sequence which is asymptotically optimal for the *Cyclic Discrete* model, and more importantly, this result can be easily extended to the general Flow Shop Weighted Completion Time Problem with any discrete distributions of the processing times and the weight. Indeed, in a companion paper, Kaminsky and Simchi-Levi (1997) show that SPT is asymptotically optimal for the equal-weight continous model under certain assumptions on the distributions of the processing times.

Table I
Uniformly Generated Computational Data

Distribution		Uniform					
Weights		Uniform			Equal		
Machines		3	6	12	3	6	12
500 Jobs	Trial 1	108%	114%	124%	109%	117%	121%
	Trial 2	107%	115%	124%	106%	115%	120%
	Trial 3	106%	113%	126%	108%	113%	118%
	Average	108%	114%	125%	108%	115%	120%
1000 Jobs	Trial 1	104%	110%	117%	105%	108%	114%
	Trial 2	109%	112%	116%	106%	111%	116%
	Trial 3	106%	110%	118%	105%	109%	116%
	Average	106%	111%	117%	105%	109%	115%
2500 Jobs	Trial 1	105%	107%	111%	103%	106%	111%
	Trial 2	103%	107%	112%	103%	105%	109%
	Trial 3	104%	107%	107%	103%	106%	110%
	Average	104%	107%	110%	103%	106%	110%
5000 Jobs	Trial 1	103%	106%	108%	102%	104%	107%
	Trial 2	102%	105%	108%	102%	103%	107%
	Trial 3	103%	105%	105%	102%	104%	109%
	Average	103%	105%	107%	101%	104%	107%

Thus, we are motivated to test the effectiveness of this rule on randomly generated problem sets.

Tables I and II both compare WSPT objective values with a lower bound for various numbers of machines. The lower bound,

$$Z^* \geqslant \frac{1}{m} Z_1^* + \frac{1}{m} \sum_{i=1}^{n} \sum_{k=2}^{m} w_i (k-1) t_i^k,$$

can easily be derived using an approach similar to the one employed for the proof of Lemma 2.1. The percentages given are the ratio of the objective value to this lower bound. For the trials described in Table I, processing times were generated from a uniform (0, 1] distribution. For those trials which utilize general job weights, the weights were also generated from a uniform (0, 1] distribution. For each combination of job number, machine number, and either general weights or equal weights, three different random trials were performed, and both individual data and averages are shown.

For the trials described in Table II, processing times were generated from an exponential distribution with mean 1. Again, for those trials which utilize general job weights, the weights were generated from a uniform (0, 1] distribution.

This limited computational testing indicates that WSPT is an effective heuristic for the Flow Shop Weighted Completion Time Problem when instances get larger and the number of machines is small. For instance, when the number of jobs increases from 500 to 5000, the relative gap goes down from about 8 percent to 3 percent for 3 machines, and from about 25 percent to 7 percent for 12

Table II
Exponentially Generated Computational Data

Distribution		Exponential					
Weights		Uniform			Equal		
Machines		3	6	12	3	6	12
500 Jobs	Trial 1	112%	120%	139%	110%	119%	133%
	Trial 2	111%	118%	139%	109%	121%	137%
	Trial 3	109%	121%	138%	110%	117%	130%
	Average	111%	120%	139%	110%	119%	133%
1000 Jobs	Trial 1	108%	113%	128%	107%	114%	120%
	Trial 2	108%	116%	123%	106%	112%	122%
	Trial 3	116%	119%	127%	106%	113%	123%
	Average	107%	116%	126%	106%	113%	122%
2500 Jobs	Trial 1	105%	108%	119%	103%	108%	115%
	Trial 2	106%	110%	120%	104%	108%	114%
	Trial 3	105%	111%	119%	105%	109%	116%
	Average	105%	110%	120%	104%	108%	115%
5000 Jobs	Trial 1	103%	107%	113%	103%	105%	111%
	Trial 2	104%	107%	113%	103%	106%	111%
	Trial 3	104%	107%	112%	104%	106%	110%
	Average	103%	107%	113%	103%	106%	111%

Table III
Computational Tests of the Modified Profile Fitting Heuristic

Distribution		Uniform							
Weights		Uniform				Equal			
Look Ahead		none	10	20	50	none	10	20	50
3 Machines	250 Jobs	112%	105%	104%	104%	112%	106%	104%	103%
	500 Jobs	109%	105%	103%	103%	107%	104%	102%	101%
	1000 Jobs	106%	103%	102%	101%	106%	104%	103%	101%
6 Machines	250 Jobs	120%	115%	114%	117%	118%	111%	108%	107%
	500 Jobs	115%	110%	109%	109%	112%	107%	106%	104%
	1000 Jobs	111%	108%	107%	105%	109%	106%	105%	104%
12 Machines	250 Jobs	133%	128%	129%	138%	128%	122%	120%	116%
	500 Jobs	124%	120%	120%	122%	121%	117%	115%	111%
	1000 Jobs	117%	115%	113%	113%	115%	112%	110%	106%

machines, when the weights are general and processing times are uniform (0, 1].

5.2. Modified Profile Fitting Heuristic

Although the computational results reported above are fairly good in many cases, the WSPT rule is not as effective when the number of machines is large and the number of jobs is small. This motivated us to modify the WSPT rule in order to achieve better results in these cases. Our objective was to develop an algorithm that is both computationally efficient and easy to implement.

For this purpose, we utilized the structural insight stemming from the analysis described in Section 3. Thus, an effective algorithm should start with the WSPT sequence and then try to shift jobs around so that the *shifted group* structure described in the upper bound of Section 3 and depicted in Figures 1 and 2 is better approximated. That is, all jobs in the same group should be processed together such that there is no machine *idle time* when these jobs are processed.

Indeed, minimizing idle time is one of the goals of Profile Fitting Heuristic (developed by McCormick et al. 1989; see also Pinedo 1995) for minimizing the *makespan* in a flow shop with *blocking*. For that reason, we adopt a version of the profile fitting strategy *modified* to fit the Flow Shop Weighted Completion Time model, and to better achieve our goals.

To effectively achieve a sequence which minimizes idle time, we utilize the concept of *job profiles* from McCormick et al. (1989). Order the jobs according to the WSPT rule and let $D_{i,j}$ be the time that job j departs from machine i, for $j = 1, 2, \ldots, n$ and $i = 1, 2, \ldots, m$. Clearly, the departure times of job 1 from each of the machines are calculated as follows:

$$D_{i,1} = \sum_{k=1}^{i} t_1^k, \quad i = 1, 2, \ldots, m.$$

Assume $D_{i,1}, D_{i,2}, \ldots, D_{i,j-1}$ have been calculated for $i = 1, 2, \ldots, m$ and some $j \geq 2$, then

$$D_{1,j} = D_{1,j-1} + t_j^1,$$

$$D_{i,j} = \max(D_{(i-1),j}, D_{i,j-1}) + t_j^i, \quad i = 2, 3, \ldots, m.$$

Therefore, the total idle time between any two consecutive jobs, $j-1$ and $j, j \geq 2$, $I_{j-1,j}$ is determined as follows:

$$I_{j-1,j} = \sum_{k=2}^{m} \{D_{k,j} - t_j^k - D_{k,j-1}\}.$$

Observe that in the optimal interchange strategy for the cyclic discrete model developed in Section 3, two consecutively scheduled jobs, u and v, that belong to the same group have $I_{u,v} = 0$. Thus, the smaller the value $I_{j-1,j}$ is, the smaller the machines idle time is, and the better jobs tend to "fit together."

These concepts are used in the following **Modified Profile Fitting Heuristic**.

Step One. Sort all jobs in a list from largest to smallest according the ratio of weight to total processing time.

Step Two. Remove the first job, job j, from the list and schedule it next.

Step Three. Search up to the first L remaining jobs in the list to find the job l which, if scheduled next, would minimize the function $I_{j,l}$. Move this job to the first position in the list.

Step Four. Go to Step Two.

The parameter L, called the look ahead parameter, clearly has an impact on both the quality of the solution and the speed of this algorithm. To determine the effect this parameter has on solution quality, we performed some limited computational testing of the **Modified Profile Fitting Heuristic**. Table III compares the ratio of the objective value to the lower bound for problems with processing times generated from a uniform (0, 1] distribution, both when the weights are all equal and when they are generated from a uniform (0, 1] distribution. Each entry represents the average of three random trials. The heuristic was tested on problem instances with several different numbers of jobs, and various different settings of the look ahead parameter L. The Look Ahead row lists the parameter L settings; "none" indicates that simple WSPT was utilized. This limited computational testing suggests that the **Modified Profile Fitting Heuristic** effectively reduces the gap

Table IV
Computational Tests of Industrial Data

Machines	2	3
Jobs	176	143
10 Job Look-Ahead	95%	98%
20 Job Look-Ahead	91%	97%
40 Job Look-Ahead	90%	94%

between the heuristic solution and lower bound when compared to WSPT, especially on the unweighted problem sets.

5.3. Industrial Data

Although we did not have any industrial flow shop data to test WSPT rule and the **Modified Profile Fitting Heuristic**, we did have some large-scale job shop industrial data. We extracted a list of jobs that went to the same two and three machines from these data along with their processing times on each of the machines, and used this as sample flow shop data. As these are relatively small data sets and our lower bound is weak for small problem instances, we compared objective values obtained using SPT (these data are unweighted) to those obtained utilizing the **Modified Profile Fitting Heuristic**. Table IV lists the ratios of objective values obtained utilizing the **Modified Profile Fitting Heuristic** with various Look Ahead parameters to the objective value obtained using SPT. The reduction in total completion time due to the **Modified Profile Fitting Heuristic** is evident.

6. DISCUSSION, EXTENSIONS, AND CONCLUDING REMARKS

The analysis performed in this paper can be carried over to several more general versions of the Flow Shop Weighted Completion Time Problem. First, consider a version of this model in which one is allowed to process jobs on different machines in different sequences, a **nonpermutation** schedule. In fact, Theorem 1.2 can be extended for this model as well, since the lower bound developed in Lemma 2.1 holds even when one allows different schedules on different machines. Thus, we can perform exactly the same analysis on the more general model to conclude that asymptotically the restriction to a single sequence does not have any impact on the optimal cost.

Next, consider a version of this model in which the intermediate storage available between successive machines is limited. Clearly, Theorem 1.2 can also be extended to this model, since in the upper bound developed in Section 4, no job ever waits for any machine other than the first one. In particular, this implies that Theorem 1.2 is valid even for a model in which there is **no** intermediate storage available between successive machines. Thus, asymptotically, the presence or absence of storage space between successive machines has no impact on optimal cost.

In addition, the analysis and results in this paper can be extended to a problem with **random routing**, provided that **each job visits each machine**. There are $m!$ possible routings in an m machine shop if each job visits each machine exactly once. Index all possible routings $1 \ldots R$, and let r_i, $i = 1 \ldots R$ be the probability that a job follows routing i. Now, consider the model described in Section 1, but extended so that rather than each job visiting each machine in the same sequence, each has routing i with probability r_i.

To provide some insight into the analysis of this extended model, consider the *Cyclic Discrete Model* developed in Section 3, extended so that each *group type* is defined by a particular routing, in addition to the time assignment vector and weight used in the original model. The lower bound developed in Lemma 2.1 still holds for this extended model. Similarly, an analogous upper bound can be constructed because each *group type* has a particular routing and in that upper bound there is no overlap between jobs in different group types. It is therefore easy to see that Theorem 3.1 is still valid for this more general case. Theorem 1.2 can be extended similarly.

Finally, this paper would be incomplete without some remarks concerning the weaknesses of our model. First, although the Weighted Completion Time Model does consider individual job related objectives, it does not consider due-date related objectives. Clearly, many real-life managerial scheduling concerns focus on job due dates and related objectives. Also, we focus on static scheduling situations where all jobs are released and available simultaneously. In actual factories, jobs are often arriving all the time, and good schedules must dynamically adjust to these new arrivals. In addition, actual processing times are often stochastic, and machines can break down, although in our model we assume that machines are always available and processing times are all known beforehand. The distribution of processing times on successive machines may not be identical or independent. Finally, this asymptotic analysis assumes very large numbers of jobs, and it can be seen from our computational work that WSPT is not very effective for small numbers of jobs. Nevertheless, it is encouraging to see that relatively complex scheduling problems are amenable to rigorous mathematical analysis leading to insights into the structure of optimal solutions to large-scale scheduling problems. We hope to build on this insight in future research as we alter our models to address some of these issues.

ACKNOWLEDGMENT

We are indebted to Professor Michael Pinedo for pointing out the connection between our work and the Profile Fitting Heuristic. Also, we would like to thank the anonymous referees and associate editor for their thorough review and valuable suggestions. Research supported in part by ONR Contracts N00014-90-J-1649 and N00014-95-1-0232, NSF Contracts DDM-9322828 and DMI-9732795, and a grant from S&C Electric Corporation.

REFERENCES

Adir, I. and N. Amit. 1984. Openshop and Flowshop Scheduling to Minimize Sum of Completion Times. *Comput. Opns. Res.* **11,** 275–284.

Bhaskaran, K. and M. Pinedo. 1992. Dispatching. In *Handbook of Industrial Engineering*. G. Salvendy (ed.), Wiley, New York.

Bramel, J. and D. Simchi-Levi. 1995. A Location Based Heuristic for General Routing Problems. *Opns. Res.* **43,** 649–660.

Bramel, J. and D. Simchi-Levi. 1996. Probabilistic Analyses and Practical Algorithms for the Vehicle Routing Problem with Time Windows. *Opns. Res.* **44,** 501–509.

Chan, L. M. A., A. Muriel, and D. Simchi-Levi. 1996. Parallel Machine Scheduling, Linear Programming and Parameter List Scheduling Heuristics. *Opns. Res.*

Coffman, E. G., G. N. Frederickson, and G. S. Lueker. 1982. Probabilistic Analysis of the LPT Processor Scheduling Heuristic. *Deterministic and Stochastic Scheduling*. M. A. H. Dempster et al. (eds.), D. Reidel Publishing Company. 319–331.

Frenk, J. B. G. and A. H. G. Rinnooy Kan. 1987. The Asymptotic Optimality of the LPT rule. *Math. O. R.* **12,** 241–254.

Garey, M. R., D. S. Johnson, and R. Sethi. 1976. The Complexity of Flowshop and Jobshop Scheduling. *Math. O. R.* **1,** 117–129.

Ignall, E. and L. Schrage. 1965. Application of the Branch and Bound Technique to some Flow-Shop Scheduling Problems. *Opns. Res.* **13,** 400–412.

Kaminsky, P. 1997. Doctoral Dissertation: Probabilistic Analysis and Practical Algorithms for Large Scale Scheduling Problems. Northwestern University.

Kaminsky, P. and D. Simchi-Levi. 1998. Shop Scheduling at S&C Electric Co., Internal Report, Northwestern University.

Kaminsky, P. and D. Simchi-Levi. 1997. The Asymptotic Optimality of the SPT Rule for the Flow Shop Mean Completion Time Problem. Working Paper, Northwestern University.

Kohler, W. and K. Steiglitz. 1975. Exact, Approximate, and Guaranteed Accuracy Algorithms for the Flow Shop Problem $n/2/F/\bar{F}$. *J. Assoc. Comput. Machinery* **22,** 106–114.

Krone, M. J. and K. Steiglitz. 1974. Heuristic Programming Solutions of a Flowshop Scheduling Problem. *Opns. Res.* **22,** 629–638.

Loulou, R. 1984. Tight Bounds and Probabilistic Analysis of Two Heuristics for Parallel Processor Scheduling. *Math. O. R.* **9,** 142–150.

McCormick, S. T., M. L. Pinedo, S. Shenker, and B. Wolf. 1989. Sequencing in an Assembly Line with Blocking to Minimize Cycle Time. *Opns. Res.* **37,** 925–936.

Morton, T. and D. Pentico. 1993. *Heuristic Scheduling Systems*. Wiley-Interscience, New York.

Pinedo, M. 1995. *Scheduling: Theory, Algorithms and Systems*. Prentice Hall, Inc., Englewood Cliffs, NJ.

Ramudhin, A, J. J. Bartholdi, J. Calvin, J. H. Vande Vate, and G. Weiss. 1996. A Probabilistic Analysis of 2-Machine Flowshops. *Opns. Res.* **44,** 899–908.

Spaccamela, A. M., W. S. Rhee, L. Stougie, and S. van de Geer. 1992. Probabilistic Analysis of the Minimum Weighted Flowtime Scheduling Problem. *O. R. Lett.* **11,** 67–71.

Szwarc, W. 1983. The Flow-Shop Problem with Mean Completion Time Criterion. *IIE Trans.* **15,** 172–176.

van de Velde, S. L. 1990. Minimizing the Sum of the Job Completion Times in the Two-Machine Flow Shop by Lagrangean Relaxation. *Ann. Opns. Res.* **26,** 257–268.

Webster, S. 1993. Bounds and Asymptotic Results for the Uniform Parallel Processor Weighted Flow Time Problem. *O. R. Lett.* **41,** 186–193.