Forecasting Waiting Times in Dynamic Stochastic Systems

Philipp H. Rossner, Oliver Holthaus
Department of Production, Operations and Logistics Management, University of Passau, Innstrasse 39, 94032 Passau, Germany, {rossne01@stud.uni-passau.de, holthaus@uni-passau.de}

W. David Kelton
Department of Quantitative Analysis and Operations Management, University of Cincinnati, PO Box 210130, Cincinnati, Ohio 45221-0130, USA, david.kelton@uc.edu

Abstract

We present a new approach for forecasting waiting times for entities moving through dynamic stochastic systems that allows for state-of-the-art look-ahead computations to forecast future waiting times better. The method is based on geometric Brownian motion and may be combined with any global dispatching rule to improve performance of dynamic systems. The classical *job-shop scheduling problem* (JSP) is considered as a benchmark to assess the effectiveness of the presented approach. A probability estimate is computed to detect how likely a given job will accumulate any more waiting time on its remaining way through the system. The proposed method is therefore called *No-Queuing probability*. A simulation study shows that the performance of global dispatching rules can be improved significantly by adding the introduced No-Queuing probability to the dispatching equation. This paper makes a first attempt at presenting a way to account for structured uncertainty in stochastic systems. We introduce stochastic processes as one way to model and finally incorporate system dynamics into the dispatching decision.

1 Introduction

The classical *job-shop scheduling problem* (JSP) involves a set of jobs, each of which requires a set of machines for a certain period of time for processing. Each job consists of a sequence of operations that represent the production steps of the jobs. Each operation needs a certain machine for a certain time called the *processing time*. Once an operation is started, no interruptions are allowed until completion, i.e., there is no preemption. Job-processing times and their corresponding machine requirements represent the *job routing*. Each machine has a capacity of one, i.e., it can process one and only one operation at a time (Kutanoglu and Wu 1999). The scheduling objective is to optimize a company-related objective function.

The three common types of factory control algorithms are *dispatching*, *scheduling*, and *pull* algorithms (Baker 1998). Since pull algorithms are triggered by the demand side, strictly speaking, they forbid the optimization of any predefined target function and delegate factory control to the market.

A scheduling algorithm for a given planning horizon determines when which jobs will use which factory resources (Baker 1998). These algorithms, also referred to as *offline scheduling* (Sabuncuoglu and Hommertzheim 1992, Sabuncuoglu and Karabuk 1998), consider a manufacturing system as a closed system with complete knowledge of all the jobs to be scheduled. Monolithic optimization models are run on a centralized processor to solve the JSP with respect to the overall system (Dewan and Joshi 2001). Scheduling jobs in manufacturing-shop floors is NP-hard (Morton and Pentico 1993, Pinedo 1995). This is true even when dealing with a system in which everything is static and deterministic (Lee et al. 1997). Hence, the solution space of real-life shop configurations does not allow calculation of optimal schedules. There have been various attempts to find near-optimal solutions (Lee et al. 1997, 2002; Rai et al. 2002). Unfortunately, all centrally derived schedules are exposed to massive perturbations. As a result of various uncertainties and dynamics present in the production system, the schedule often becomes obsolete almost the moment it is released for execution (Wu et al. 1999). All together, scheduling can be very tedious with offline methods because of the difficulty of generating the schedule and updating it in a dynamic manufacturing environment (Stecke and Solberg 1981).

Dispatching algorithms tackle the problem much differently. A dispatching rule, in general, is used to select the next job from a set of waiting jobs to be processed at a machine when the machine becomes idle (Jayamohan and Rajendran 2000). Because scheduling decisions are obviously delayed until the last moment, dispatching algorithms are also referred to as *real-time scheduling* (Hutchinson 1991), *online scheduling* (Sabuncuoglu and Hommertzheim 1992, Sabuncuoglu and Karabuk 1998), or *ad hoc scheduling* (Kutanoglu and Wu 1999). Obviously, dispatching rules distribute shop-floor control equally across the various subunits of a manufacturing system and, thus, achieve control-system robustness through fault-tolerance ability of the control architecture (Duffie and Piper 1986, 1987; Duffie et al. 1988; Duffie 1990; Duffie and Prabhu 1994; Siwamogsatham and Saygin 2004).

This paper concentrates on the use of dispatching algorithms to improve job-shop performance. The objective is to minimize the mean flow time across all jobs.

2 Literature Review

Motivated by the promise of improved reliability and extensibility, as well as the potential for increased tolerance to uncertainty in data and knowledge (Smith and Davis 1981, Davis and Smith 1983, Malone et al. 1983, Shaw and Whinston 1985, Shaw 1988, Duffie 1990, Upton et al. 1991, Lin and Solberg 1992), leading US manufacturers and government agencies are claiming now that agile manufacturing is the future (Baker 1998). In a distributed system there is no organizational hierarchy—the structure is flat with a single level of control with each control point in a level operating autonomously and communicating in a peer-to-peer relationship (Dewan and Joshi 2002). Distributed scheduling is characterized by a collective rather than centralized approach to decision making with loosely coupled decentralized problem solvers motivated by their local objectives and constraints (Dewan and Joshi 2001). Although distribution of information and decision-making responsibility through implementation of local dispatching policies leads to system robustness (Duffie 1990, Valkenaer et al. 1994), distributed scheduling is myopic and the quality of the resulting solutions may not be as good as in the case of centralized methods (Sabuncuoglu and Karabuk 1998). Consequently, we are interested in mechanisms that allow resource scheduling to be locally autonomous yet also aligned with global interests (Kutanoglu and Wu 1999). More sophisticated models involving the co-operative solution of problems by a set of decentralized and loosely connected intelligent problem-solving agents point in that direction (Suresh and Chaudhuri 1993, Pechoucek et al. 2002, Dewan and Joshi 2002, Babayan and He 2004, Siwamogsatham and Saygin 2004, Srinivas et al. 2004).

However, global dispatching algorithms are yet a more sophisticated way to increase the quality of dispatching decisions from an overall perspective. While the scheduling decisions are still made autonomously by every part of the system, global knowledge, such as the average waiting time at remote parts of the system, is considered. In general, dispatching rules can be classified into two broad categories: *local policies*, which use information pertaining to the immediate neighborhood of the decision point in space and time, and *global dispatching rules*, which incorporate additional information from remote parts of the production system. The latter have proven more effective as they do a better job in balancing the line output while achieving the performance measure of interest (Dabbas and Fowler 2003). Various researchers present rich reports on most of the dispatching rules being used for job-shop scheduling (Conway 1965, Conway et al. 1967, Panwalkar and Iskander 1977, Blackstone et al. 1982, Haupt 1989). Since dispatching rules are normally intended to minimize work-in-process inventory or tardiness costs, it is a customary practice to minimize flow-time-related and tardiness-related measures of performance since the associated inventory and tardiness costs are assumed to be directly proportional to the time periods of flow time and tardiness of jobs (Blackstone et al. 1982). So far, no rule has been found to perform uniformly well for all criteria relating to flow times and tardiness. The SPT rule is still considered to be the most effective local dispatching rule in minimizing mean flow times (Dabbas and Fowler 2003). In general, it has been noted that process-time-based rules fare better under tight load conditions, while due-date-based rules perform better under light load conditions (Conway 1965, Rochette and Sadowski 1976). Recently, many researchers have proposed global dispatching rules that outperform SPT in terms of flow-time minimization (Raghu and Rajendran 1993; Lu et al. 1994; Lin 1996; Holthaus and Rajendran 1999, 2000; Hung and Chang 2002). The basic intuition underlying highly effective dispatching rules is to forecast future system states in order to determine the expected waiting time of the various jobs.

3 Waiting-Time Prediction

If there were a way to get a good picture of future system status, we would just have to dispatch those jobs that are most likely to accumulate the least additional waiting time on their remaining way through the system in order to minimize mean flow times across all jobs. Refer to the term that computes an estimate of the future system status as the *look-ahead component* of a dispatching rule. The simplest way to formulate such a look-ahead component is to follow a martingale approach: if the system is not expected to change over the short run, the future system status is best represented by the current system status. Raghu and Rajendran (1993) as well as Holthaus and Rajendran (1999) proposed a look-ahead component that relies on such intuition. The latter used the total work stored in the next queue (WINQ) as an estimate of the future waiting time of a given job. The proposed dispatching rule is global by nature and is very effective in terms of minimizing mean flow times. Although it requires minimum computational effort, it might lead to suboptimal results, since production systems are exposed to perturbations so that a martingale approach might no longer be appropriate. Lu et al. (1994), Lin (1996), and Hung and Chang (1999, 2002) rely on an iterative simulation approach, exponential smoothing, and an empirical queuing approach, respectively, to predict future waiting times efficiently and accurately.

We share common ground with previous research since we assume that forecasting waiting times allows for more effective dispatching in a distributed-scheduling environment.

3.1 Nature of Volatility

We assume that the transportation times and set-up times within a production side are trivial, the inter-arrival times between jobs are exponentially distributed, and the processing times for every work step within the machining sequence of a given job are drawn from a uniform distribution, and that we know when the job is released to the shop floor.

Like many other social systems, a shop floor is heavily related to various factors in the environment. Even if utilization rates stay stable, a change in factors such as the product mix, item due-dates, quality-control procedures, priority listings, or lot sizes challenge shop-floor control. Indeed, all aforementioned factors lie within the control of the company's management. However, there are enough external factors such as traffic, weather, customer demand, competition, unions, and macroeconomic variables that account for variation in the number and the kind of jobs that are released into the shop floor. Although it is impossible for management to control such external factors, it is reasonable to assume that they are independent from each other and, thus, can be described as idiosyncratic risk. The cumulative impact of all different factors should then oscillate around a basic trend, and we can expect the drift and volatility parameters of various system ratios, such as the average number of waiting jobs, to stay stable over the short run.

3.2 Probabilistic Evaluation of Future System States

We introduce a general procedure to compute the look-ahead component to improve forecasts of future waiting times. We rely on stochastic processes as a means of calculating probabilities for various system changes. Our intent is to enrich the dispatching decision through a probability assessment that answers the following question: how likely is it for a job to be processed at its next operation without accumulating additional waiting time, given that it is chosen to be sent for processing at its current work step? Let us call this the *No-Queuing probability* P_j of job j. A dispatching rule should then choose the job with the highest No-Queuing probability P_j.

Consider a job shop consisting of M machines, where every machine makes its own dispatching decision based on the SPT rule. Pick machine m_1 and assume that there are two jobs j_i, $i = 1, 2$, waiting for processing. Let ρ_{jm} denote the process time (PT) of job

Table 1: Process Times in Minutes

Job j_i	Machine m_i		
i	1	2	3
1	10	35	–
2	20	–	50

j at machine m and let both jobs be different and require different machining sequences: j_1 has to visit m_2 next while j_2 has to visit m_3 next. Suppose, for example, that there are currently 5 jobs waiting at machine m_2 and that the process times across those jobs are normally distributed with a mean of 20 and a standard deviation of 5 minutes. On the other hand, there are 50 jobs waiting at machine m_3, and the process times for these jobs is normally distributed with a mean value of 200 and a standard deviation of 50 minutes. We acknowledge that process times cannot be normally distributed in the sense that they would then have a chance, however small, to take on negative values. However, the simulation results in section 6 appear to be robust to departures from this normality assumption. Now, given Table 1 for the processing times ρ_{jm}, which job should be chosen for processing at machine m_1?

SPT requires the workstation m_1 to choose job 1. However, given the current situation at its subsequent workstation and given that the SPT policy is implemented at every part of the system, it is likely for job 1 to accumulate more than 100 minutes of additional waiting time at machine m_2. At the same time, job 2 is likely to be processed instantly without accumulating any more waiting time at its subsequent workstation m_3, since its process time at this machine ρ_{23} lies more than 3σ below the average process time of all jobs currently waiting at m_3. Clearly, the dispatching decision made solely based on SPT should be revised through the consideration of the jobs' No-Queuing probabilities. It is much more likely for job j_2 to be processed prior to all other jobs waiting at machine m_3 than it is for job j_1 at machine m_2. Up to now, we have considered a static environment in that the system status does not change over time. Now, let us go one step further and introduce system dynamics. If we assume that the process underlying the average process time of all jobs waiting at any machine m follows geometric Brownian motion, we can make a more sophisticated dispatching decision. Should the $N(20, 5)$ distribution of the process times at machine m_2 represent an all-time low and start to move upward over time, then our previous dispatching decision "choose job 1 for processing" might not continue to be an optimal one.

The picture gets even more complex if we consider the volatility of the diffusion process. The remainder of this paper introduces an approach to forecast waiting times based on stochastic processes that considers all of the aforementioned factors.

4 No-Queuing Probability P_j

There are multiple ways to measure the introduced No-Queuing Probability P_j for a given job j. We measure P_j as the probability that the mean process time of the jobs waiting at the subsequent machine m' in the machining sequence of job j falls below the process time $\rho_{jm'}$ of job j at machine m' by the time it gets there. Since the dispatching decision at the current machine m is to be made at time t_0, it takes a given job ρ_{jm} time units to get to machine m'. Let $t' = t_0 + \rho_{jm}$ represent the moment when job j arrives at machine m' (remember that we have assumed transportation times to be 0). Now, we can rewrite the No-Queuing probability as $P_j(\bar{\rho}_{m't'} < \rho_{jm'} \mid \bar{\rho}_{m't_0})$ where $\bar{\rho}_{m't'}$ denotes the mean process time across all jobs waiting at machine m' at time t'. m' simply denotes the next work station in the machining sequence of a given job. Let us develop a model that calculates P_j exactly.

4.1 Geometric Brownian Motion as the Underlying Process

As mentioned earlier, many social systems are exposed to various independent risks that sum up to a cumulative risk factor that oscillates around a basic trend over the short run. Although the impact of the various risk factors changes on a continuous basis, we expect the cumulative impact to follow stable patterns over the short run.

Stochastic processes can be described by a drift and a volatility parameter. In the special case of production systems, many ratios such as the average process time across all jobs waiting at a particular machine must be positive numbers. Therefore, we use geometric Brownian motion as the underlying process to model the development of $\bar{\rho}_{mt}$ over time. Rossner (2005) gives statistical evidence that $\bar{\rho}_{mt}$ follows geometric Brownian motion for a specified, and highly loaded production system. The process that mimics the development of $\bar{\rho}_{mt}$ at machine m is denoted as $d\theta_{mt}$. Clearly, the stochastic process is not able to predict future values for $\bar{\rho}_{mt}$. However, the trend and volatility parameters that underlie both time series have to be identical. Hence, the probability that the virtual random walk $d\theta_{mt}$ undergoes a certain value $x*$ in a finite time window $[t_0, t']$ should equal the probability that the real process $\bar{\rho}_{mt}$ undergoes the same value $x*$ in $[t_0, t']$. Let us therefore derive P_j

from the stochastic differential equation of a geometric Brownian process. We will see later what limits the use of a geometric Brownian motion. The use of more complex processes such as *jump-diffusion* movements might be more appropriate to capture the various kinds of underlying risk factors. Since geometric Brownian motion does a good job in most situations and since it is easy to implement, we adopt it for our purposes. For the remainder of the paper, we refer to $d\theta_t$ as the stochastic processes without specifying a specific machine m.

4.1.1 Derivation of P_j

This section requires an understanding of stochastic differential equations; see Oksendal (1998) for background. Write $d\theta_t$ as a stochastic process in continuous time as

$$\frac{d\theta_t}{dt} = a(t)\theta_t \tag{1}$$

where $a(t) = \alpha + \sigma \times W_t$. W_t specifies a *Wiener process*—also referred to as standardized Brownian motion—that has drift and volatility parameters of 0 and 1, respectively. W_t can be described as a martingale. We can rewrite (1) as

$$d\theta_t = \alpha\theta_t dt + \sigma dW_t \theta_t \tag{2}$$

where the drift parameter α specifies the expected growth, and the volatility σ describes the mean relative deviation of the process from α. The Wiener process dW_t accounts for the randomness in the model and can be expressed as

$$dW_t = \varepsilon\sqrt{dt} \tag{3}$$

where ε is a $N(0,1)$ distributed random variable. By the central limit theorem, the ending values of a random walk described by a Wiener process are normally distributed, since they simply represent the sum of various identically distributed random variables. This result will help us to derive the distribution for P_j.

Assuming that this model fully describes the fluctuations of $\bar{\rho}_{mt}$ at any machine m over time, we can use Ito's lemma to solve the stochastic differential equation (2).

Theorem 1 *(Ito 1951). Let $dx_t = \alpha dt + \sigma dW_t$ be a stochastic process with a drift rate α and volatility σ. The increments in continuous time of a twice-differentiable function $G(t, x)$ can be written as*

$$dG = \left[\frac{\delta G}{\delta t} + \frac{\delta G}{\delta x}\alpha + \frac{1}{2}\sigma^2\frac{\delta^2 G}{\delta x^2}\right] dt + \frac{\delta G}{\delta x}\sigma dW_t.$$

dG then describes the process as dx_t with identical values for α and σ.

For a function $f(t, \theta_t) = \ln \theta_t$ we can write

$$\frac{\delta f}{\delta \theta_t} = \frac{1}{\theta_t}, \qquad \frac{\delta^2 f}{\delta \theta_t^2} = -\frac{1}{\theta_t^2}, \qquad \text{and} \qquad \frac{\delta f}{\delta t} = 0,$$

and because of *Theorem 1*

$$df = \left[0 + \frac{1}{\theta_t}\alpha\theta_t + \frac{1}{2}\left(-\frac{1}{\theta_t^2}\right)\sigma^2\theta_t^2\right] dt + \frac{1}{\theta_t}\sigma\theta_t dW_t,$$

or

$$df = \left[\alpha - \frac{\sigma^2}{2}\right] dt + \sigma dW_t, \tag{4}$$

which then describes the diffusion of $\ln \theta_t$ over time with identical values for α and σ from the underlying process (2). We can finally rewrite df as

$$df = d\ln\theta_t = \ln\theta_{t+dt} - \ln\theta_t = \ln\frac{\theta_{t+dt}}{\theta_t}.$$

Because α and σ are constant, (4) follows a Wiener process with drift. The drift rate is perfectly described by the first term in (4), while the variance is simply $\sigma^2 dt$; see (3) for the variance of a Wiener process. Since we know the distributional properties of a Wiener process, we can derive the distributional properties of the process (2) as *Theorem 2*.

Theorem 2 *The increments of the process described in formula (2) have the following properties:*

1. $\ln\frac{\theta_{t+dt}}{\theta_t} \sim N\left(\left(\alpha - \frac{\sigma^2}{2}\right)dt, \sigma^2 dt\right)$,
2. *The increments are stationary,*
3. *The increments are independent of each other.*

The first item in *Theorem 2* allows us to calculate the No-Queuing probability P_j as a function of the drift and volatility of the underlying stochastic process as well as from the remaining time dt. As mentioned earlier, dt should be taken as ρ_{jm}, the process time of job j at the current work station m.

Recall that P_j, the probability that the average process time across all jobs waiting at the next machine of job j falls below the process time of job j at machine m', given the current average process time across all jobs waiting at machine m', was $P_j(\bar{\rho}_{m't'} < \rho_{jm'} \mid \bar{\rho}_{m't_0})$ where t' is the moment when job j is expected to arrive at machine m, $t' = t_0 + \rho_{jm}$, and $t' > t_0$. Thanks to the first item in *Theorem 2* we now have a better idea of what P_j actually means: P_j describes how likely a certain movement $d\theta_t$ will occur in a given time window $[t_0; t']$. Since we already know that $d\theta_t$ is normally distributed, we can finally calculate P_j.

Let b_j denote the critical movement for which we want to calculate a probability P_j. Clearly,

$$b_j = \ln \frac{\rho_{jm'}}{\theta_{m't_0}}$$

and therefore, we can write P_j as

$$P_j(\bar{\rho}_{m't'} < \rho_{jm'} \mid \bar{\rho}_{m't_0}) = \begin{cases} P(b_j \geq a) & = & 1 - \Phi(a), & \forall b_j \geq 0 \\ P(b_j < a) & = & \Phi(a), & \forall b_j < 0 \end{cases}$$

where

$$a = \frac{b_j - (\alpha - \sigma^2/2)\, dt}{\sqrt{\sigma^2 dt}}$$

and Φ is the cumulative distribution function of the standard normal distribution.

4.1.2 Properties of the Underlying Process $d\theta_t$

We can plug (4) into the exponential function and get

$$\theta_t = \theta_0 e^{\left(\alpha - \sigma^2/2\right) dt + \sigma\varepsilon\sqrt{dt}}. \tag{5}$$

From (5) we can derive helpful insights into the implied oscillation of the stochastic process.

Theorem 3 *The stochastic process describing the diffusion of $d\theta_t$ has the following properties:*

1. *$\theta_t \to \infty$ if $t \to \infty$, for $\alpha > \sigma^2/2$,*

2. *$\theta_t \to 0$ if $t \to \infty$, for $\alpha < \sigma^2/2$,*

3. *θ_t oscillates between arbitrary small and large values for $\alpha = \sigma^2/2$.*

Since the drift rate is expected to be 1 in a steady-state system, θ_t approaches 0 when the forecast horizon is only high enough. Hence, for jobs with long process times ρ_{jm}, $t' = t_0 + \rho_{jm}$ becomes large and, thus, P_j increases. This makes perfect sense: if a given job requires long process times at the current work step, the mean process time at the next machine $\bar{\rho}_{m'}$ has more time to realize the critical movement b_j.

4.1.3 Distribution Parameters for θ_t

θ_{tm} is a function of the number of jobs n_{tm} at a given machine and the process times ρ_{jm} required by each of those jobs $j = 1, ..., n_{tm}$. Since both n_{tm} and ρ_{jm} are random variables, θ_{tm} is also a random variable. Let $N_{tm} = \{j_1, ..., j_n\}$ denote the set of jobs waiting at machine m in time t. The expected value for θ_{tm} can be written as

$$E\left[\theta_{tm}\right] = \frac{1}{n_{tm}} \sum_{i \in N_{tm}} \rho_{im}$$

and, thus, $E\left[\theta_{tm}\right] \to E\left[\rho_{jm}\right]$ if $n_{tm} \to \infty$. More interesting is the second moment of the distribution. Let ρ_{jm} be a discrete uniform random variable in the set $\{a, a+1, a+2, ..., b\}$. For $0 < a < b$, we get

$$P\left(\theta_{tm} = a \mid N_{tm} = 1\right) > P\left(\theta_{tm} = a \mid N_{tm} = n\right)$$

for every $n > 1$, since

$$P\left(a \mid 1\right) = \frac{1}{b-a} > \left(b-a\right)^{-n} = P\left(a \mid n\right).$$

Clearly, for every point in time where $n_{tm} = 1$, the distribution of θ_{tm} is discrete uniform in the interval $[a, b]$ and approaches a normal distribution as n_{tm} increases. The conditional variance $\sigma_{\bar{\rho}}\left(\theta_{tm} \mid n_{tm}\right)$ is expected to increase as n_{tm} decreases.

We conclude that the No-Queuing probability is suitable only for highly loaded systems. In lightly loaded systems, n_{tm}, the number of waiting jobs at a machine m, is likely to be low and, thus, as seen above, the mean process time across all waiting jobs tends to fluctuate widely. Based upon the implied oscillation properties of geometric Brownian motion, increased volatility $\sigma_{\bar{\rho}}$ leads to higher values for the No-Queuing probability values P_j. But as soon as P_j approaches 1 for all jobs waiting in line, the concept of No-Queuing probability is not of much help for the dispatching decision.

This result is very important for future use of the presented concept. We stress that the use of P_j is most effective in highly loaded systems and not of additional value in lightly loaded environments.

5 Implementation of the No-Queuing Probability

The remainder of this paper describes a real implementation of a dispatching rule based on the No-Queuing probability concept. We performed a simulation study to show that the proposed No-Queuing probability increases the effectiveness of any global dispatching rule in terms of minimized mean flow times. The results support the theoretical finding that the No-Queuing probability is most effective in highly loaded systems, while it does not lead to more effective dispatching decisions in lightly loaded environments.

5.1 New Dispatching Rule Based on the No-Queuing Probability

The No-Queuing probability is not to be seen as a proper dispatching rule itself, but rather as a very powerful look-ahead term that can be combined with any other global dispatching rule with respect to the minimization of mean flow time. For the simulation study, we chose three dispatching rules that have been proven very effective and integrated the No-Queuing probability into each of them. We do not provide a detailed description of the dispatching rules being used in the simulation study, but the interested reader could refer to Raghu and Rajendran (1993) and Holthaus and Rajendran (1999, 2000).

Raghu and Rajendran (1993) developed a dispatching rule on the finding that SPT achieves good results under heavy load conditions while due-date-based rules such as S/OPN perform better under light loads. Their proposed rule uses the utilization rate observed for a machine m in the past, denoted by ν_m, $0 < \nu_m < 1$. When ν_m is high the rule resembles SPT. When ν_m is low, more weight is given to a due-date-based dispatching decision. Since the rule is widely known as RR, we refer to $RR \otimes P_j$ as the *enhanced* version. $RR \otimes P_j$ can be written as

$$Z_j = \frac{Slack_j \otimes e^{-\nu_m} \rho_{jm}}{RPT_j} + e^{\nu_m} \rho_{jm} + WINQ_j \otimes P_j \left(\bar{\rho}_{m't'} < \rho_{jm'} \mid \bar{\rho}_{m't_0}\right)$$

where $Slack_j$ means the maximum time job j could spend waiting before it overshoots its assigned due date. RPT_j denotes the sum of process times of uncompleted operations (remaining process time). $WINQ_j$ indicates the already-introduced concept of work in next queue, which is the sum of process times of all jobs waiting at the subsequent machine of job j. The No-Queuing probability was incorporated in the last term of the RR rule.

Z_j is the *priority index* for job j. The job with the smallest priority index Z_j is chosen for processing.

Holthaus and Rajendran (1999) showed that $PT + WINQ$ outperforms RR in job shops with missing operations and, thus, was the most effective dispatching rule at the time. We stretch the rule to $PT + WINQ \otimes P_j$ again by incorporating the No-Queuing Probability.

$PT + WINQ \otimes P_j$ can be written as

$$Z_j = \rho_{jm} + WINQ_j \otimes P_j \left(\bar{\rho}_{m't'} < \rho_{jm'} \mid \bar{\rho}_{m't_0} \right).$$

$2PT + WINQ + NPT$ (Holthaus and Rajendran 2000) was shown to outperform $PT + WINQ$. Again, this rule is extended to $2PT + WINQ \otimes P_j + NPT$ so that the No-Queuing probability is considered as the look-ahead component. $2PT + WINQ \otimes P_j + NPT$ can be written as

$$Z_j = 2\rho_{jm} + WINQ_j \otimes P_j \left(\bar{\rho}_{m't'} < \rho_{jm'} \mid \bar{\rho}_{m't_0} \right) + NPT_j$$

where the middle term was modified by the No-Queuing probability and the last term indicates the job's process time at its next operation.

As shown below, the use of the No-Queuing probability P_j makes each of the rules above more effective.

5.2 Estimating Core Parameters

Computation of P_j requires the estimation of α and σ, and the derivation of the function $\Phi(z)$.

α and σ are time sensitive since either one depends on realizations for the random variable $\bar{\rho}_{mt}$. At a first glance it seems appropriate to estimate both parameters "on-the-fly." However, the distributional properties of the underlying stochastic process described in *Theorem 2* and *Theorem 3* do not allow for parameter estimates in continuous time. Recall that the second moment of $\bar{\rho}_{mt}$ relies on n_{mt} so that the variance of $\bar{\rho}_{mt}$ is high when n_{tm} is low and *vice versa*. If there are only a few jobs waiting, $\sigma_{\bar{\rho}}$ increases and it follows that P_j approaches one. In this case, the dispatching decision gives less weight to ρ_{jm} since high values for $\sigma_{\bar{\rho}}$ enable the underlying stochastic process to realize almost every critical movement b_j in any time. Hence, a job whose next workstation is frequented by not too many jobs has a small chance of being chosen for processing at its current work step. The easiest way to cope with obviously wrong dispatching decisions is to estimate α and σ in advance instead of "on-the-fly."

As soon as the system reaches its steady state, α approaches one and was therefore fixed to one. Based on output data from the simulation study described in section 5.3, $\sigma_{\bar{\rho}}$ was estimated as 34. The performance of the No-Queuing probability is sensitive to changes in both parameters.

There are multiple ways to implement a function $\Phi(z)$, e.g., a *Taylor-Series* expansion. We simply did a nine-times polynomial approximation of 10,000 data points derived from standard normal tables. The polynomial approximation draws the function $\hat{\Phi}(z)$ for $-5 < z < 5$. See Rossner (2005) for a detailed description of the estimated function $\hat{\Phi}(z)$.

5.3 Simulation study in SIMAN

5.3.1 Experimental Design for the Simulation Study

A job shop could be classified into an open shop and a closed shop, depending on the way in which jobs are routed in the shop. In a closed shop, the number of routings available to a job is fixed and an arriving job can follow one of the available routings. In an open shop, there is no limitation on the routing of a job and each job could have a different routing. In this paper we consider the open-shop configuration. The typical standard assumptions, such as the processing of only one operation on a given machine at a given instant, no job preemption, an operation of any job to be performed after the completion of all its previous operations, machines being the only limiting resources, no machine breakdowns, no assembly of jobs, and no parallel machines (Haupt 1989, Ramasesh 1990), are also made in this study. We assume the presence of ten machines.

To determine the machining sequence of a newly arrived job, a random permutation of a subset of the ten machines is chosen and the entering job undergoes processing in the chosen permutation order. Each machine in the chosen subset is visited exactly once. First, the number of operations for an entering job is randomly and uniformly sampled from $\{2, 3, ..., 9, 10\}$ and the corresponding machine visitations are randomly generated with no machine being revisited. For example, if the number of operations for an entering job is 6, we sample (without replacement) six different machines out of the ten to be visited by the job, say, $4-1-7-10-8-3$. In all experimental setups, the process times are drawn from a uniform discrete distribution on $\{1, 2, ..., 49\}$.

Job arrival times are generated using an exponential distribution for inter-arrival times. Two machine-utilization levels U_g are tested in the experiments, 80% and 95%. Thus, in all, there are two different utilization levels (i.e. two different mean inter-arrival times) and 15 different dispatching rules, making a total of 30 scenarios.

It is a customary for researchers to conduct simulation experiments with different parameter settings. We find in the literature that the utilization levels in the range 80%-95% are

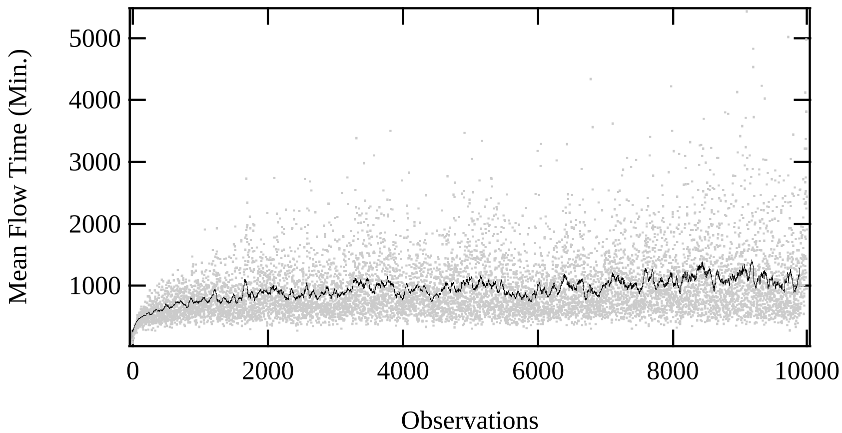

Figure 1: Mean Flow Time Across 20 Replications: Light Dots are Individual Observations, Dark Line Plots Moving Averages for 50 Observations

commonly considered (Blackstone et al. 1982, Haupt 1989). While the number of machines in a jobshop could be theoretically anything, it is usually set in the range 6-12. This setting follows from the observations of Baker and Dzielinski (1960) and Nanot (1963) that the shop size is not a significant factor in the relative performance of rules, and that a shop with about nine machines should adequately represent the complexity involved in large dynamic job-shop operations.

In our study, we are interested in steady-state or long-run performance, and each simulation experiment consists of 20 different runs (or replications). In each run, the shop is continuously loaded with job orders that are numbered on arrival. In order to ascertain when the system effectively reaches steady state, we observed the shop parameters, such as utilization level of machines, mean flow time of jobs, etc. We found that the shop appeared to reach steady state after the arrival of about 2,000 job orders (see Figure 1). Figures 1–3 plot the convergence of average flow time, number of entities in the system, and utilization rate. While the average flow time is a stochastic process in discrete time, the number of entities in the system as well as the utilization rate are both in continuous time. Since the system was warmed up in discrete time specified by number of job completions, all other variables were converted into discrete time according to the mean inter-arrival times of jobs. According to the aforementioned model settings, we need a mean inter-arrival time of approximately 15.8 minutes to satisfy a 95% utilization rate. Hence, on average every 15.8 minutes one job gets completed. This is how we convert continuous time into discrete time (see Figures 2 and 3).

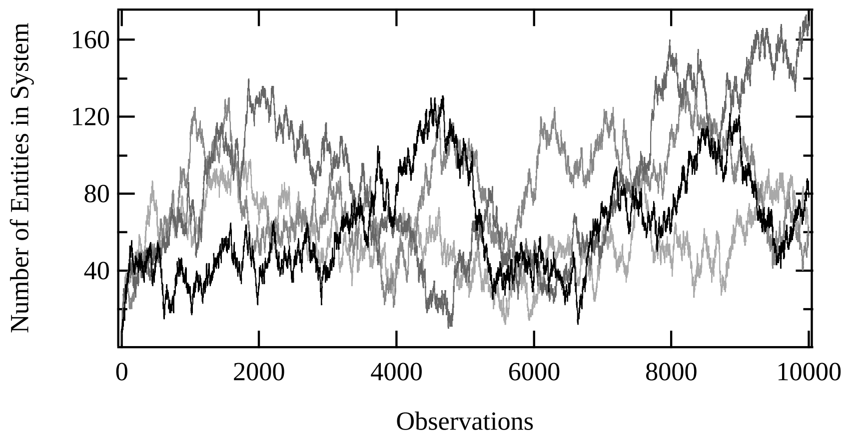

Figure 2: Number of Entities in the System in Discrete Time: Different Line Shadings Correspond to Four Samples from Independent Runs

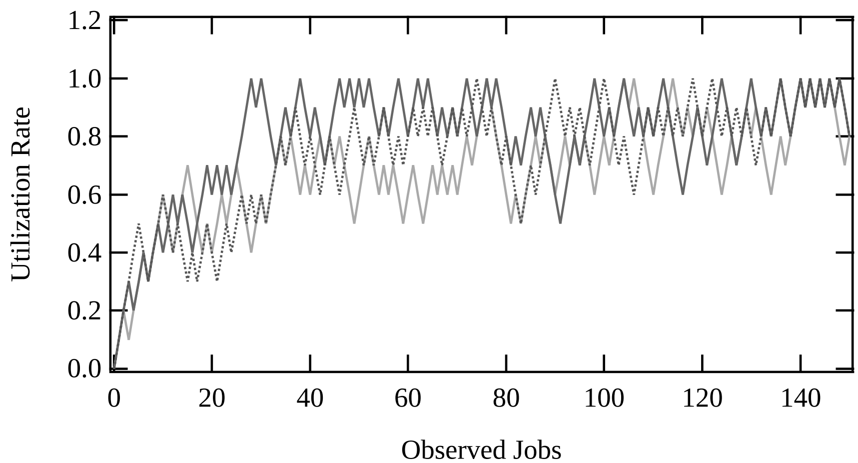

Figure 3: Utilization Rate in Discrete Time: Different Line Shadings and Patterns Correspond to Three Samples from Independent Runs

Obviously, the average flow time takes the longest to achieve steady-state. Once the average flow time has reached steady-state, both, the number of entities in the system and the utilization rate already have. Thus, we specify the the warm-up phase according to the behavior of the average flow time (see Figure 1). According to Welch (1981), "eyeballing" multi-replication plots and being conservative in the sense of warming up longer than the bare minimum indicated by the plots gives a reliable warm-up phase.

Typically, the total run length in simulation studies of jobshop scheduling is of the order of thousands of job completions (Conway et al. 1960, Blackstone et al. 1982). For a given total run length, it is preferable to have fewer replications and a longer run length, and the recommended minimum number of replications is about 10 (Law and Kelton 1984). Following these guidelines, we fixed the number of replications as 20, with the run length for every replication as 60,000 completed job orders, which provided for adequate precision to draw conclusions. As for the computation of statistics for a given replication, we collected data from orders 2,001 through 62,000, and the shop is further loaded with jobs until the completion of 64,000 job orders. This helps in overcoming the problem of "censored data" (Conway 1965). The use of very long run length produced statistically significant results in terms of small confidence intervals.

The simulation program was written in SIMAN and implemented on a Intel Pentium 2 GHz PC. See Tables 2 and 3 for the results, as well as Figures 4 and 5.

6 Conclusion

As expected by the theoretical derivation of the new concept of a No-Queuing probability, the implementation requires highly loaded systems. For systems operating under only light loads, the new concept does not improve job-shop performance for the reasons we have cited above.

Table 2 and Figure 4 show that the introduced No-Queueing probability improves the performance significantly for two of the three dispatching rules that have been tested. In case of the rule RR, incorporating the No-Queuing probability decreases the mean flow time by almost 15%. For the rule $PT + WINQ$, the performance gain is about 10%. In either case, the performance improves significantly. Only in case of the rule $2PT + WINQ + NPT$, the performance gain is not significant for a confidence level of 95%. We conclude that the randomness in the model accounts for the improvement of $2PT + WINQ \otimes P + NPT$ over

Table 2: Results after 20 Replications, 60,000 Observed Jobs per Replication, 95% Utilization Rate

Dispatching Rule	Mean Flow Time (Min.)	Half-Width (95% Confidence)
$RR \otimes P$	870.6	23.3
$2PT + WINQ \otimes P + NPT$	878.5	23.7
$PT + WINQ \otimes P$	885.0	24.1
$2PT + WINQ + NPT$	907.7	25.2
$PT + WINQ$	975.3	27.6
RR	1018.2	35.4
SPT	1019.8	31.9
$PT + WINQ/TIS$	1090.5	34.2
PT/TIS	1206.5	39.7
$PT + WINQ + SL$	1248.6	43.6
$PT + WINQ + AT$	1343.3	42.6
EDD	1542.7	53.5
AT	1565.7	53.7
$AT - RPT$	1568.9	52.8
S/OPN	1705.0	63.7

Table 3: Results after 20 Replications, 60,000 Observed Jobs per Replication, 80% Utilization Rate

Dispatching Rule	Mean Flow Time (Min.)	Half-Width (95% Confidence)
$RR \otimes P$	372.0	2.0
$2PT + WINQ \otimes P + NPT$	376.0	2.0
$PT + WINQ \otimes P$	378.0	2.2
$2PT + WINQ + NPT$	377.4	2.0
$PT + WINQ$	383.0	2.1
RR	378.0	2
SPT	386.2	2.3
$PT + WINQ/TIS$	410.4	2.6
PT/TIS	424.0	2.7
$PT + WINQ + SL$	392.0	2.7
$PT + WINQ + AT$	461.4	3.4
EDD	481.9	4.2
AT	493.8	4.2
$AT - RPT$	497.0	3.8
S/OPN	476.0	4.0

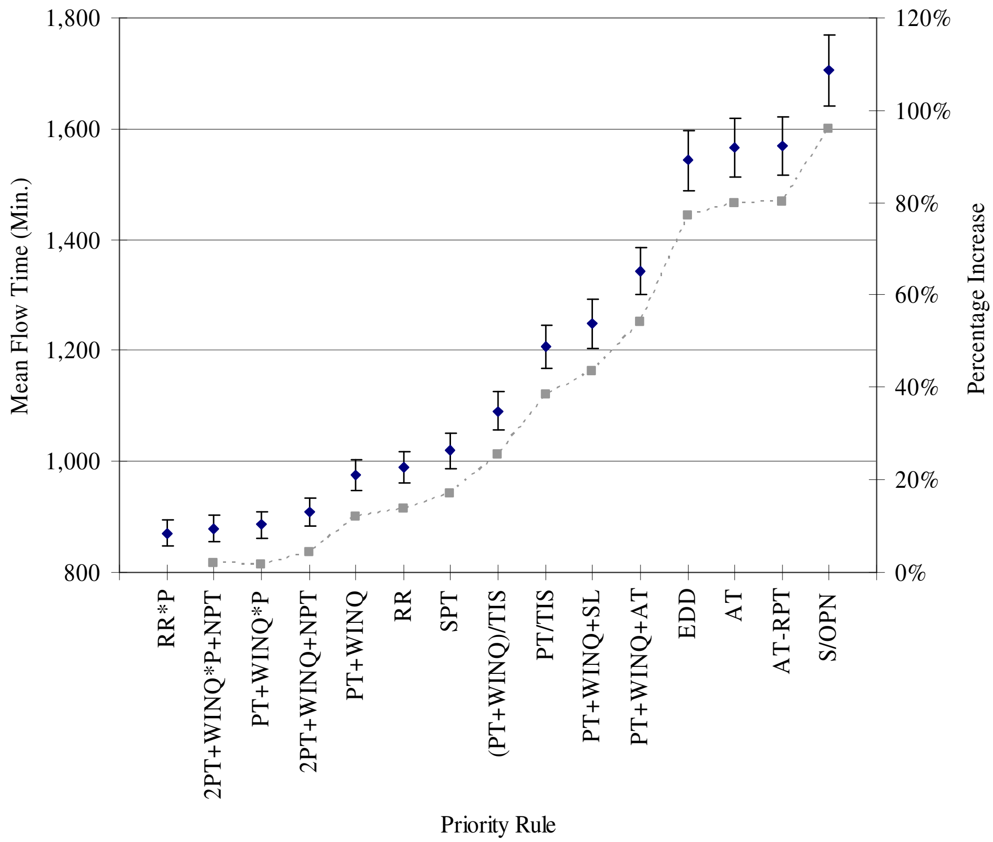

Figure 4: Results and 95% Confidence Intervals for 95% Utilization Rate: The Dotted Line Gives the Percentage Increase Over the Rule $RR \otimes P$

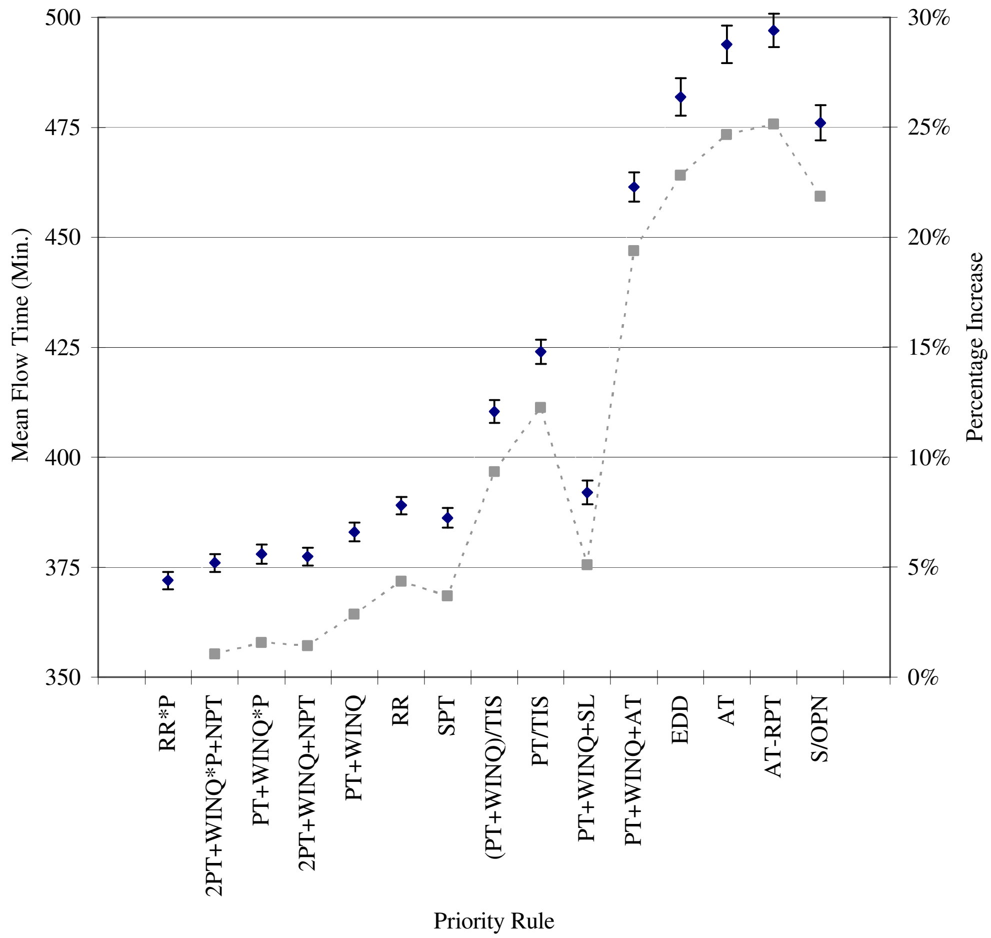

Figure 5: Results and 95% Confidence Intervals for 80% Utilization Rate: The Dotted Line Gives the Percentage Increase Over the Rule $RR \otimes P$

$2PT + WINQ + NPT$.

Table 3 shows the results for an 80% utilization rate. Even though the performance improvement of $RR \otimes P$ and $PT + WINQ \otimes P$ over RR and $PT + WINQ$, respectively, is still significant for $\alpha = 0.05$, it is expected to be not significant for every confidence level $\alpha < 0.05$.

The simulation results indicate that the introduced No-Queuing probability improves the performance of global dispatching rules in general. For low load factors, more sophisticated processes such as jump-diffusion models seem to be more appropriate to capture the erratic movements that result from the fact that random variables do not approach a normal distribution. We conclude that the No-Queueing probability should be used in combination with various global dispatching rules to improve performance in highly loaded job-shops.

This paper makes a first attempt at presenting a way to account for structured uncertainty in stochastic systems. We introduced stochastic processes as one way to model and finally incorporate system dynamics into the dispatching decision. Further research should concentrate on more sophisticated stochastic processes to model chaotic behavior that cannot be described by normally distributed increments.

References

Babayan, A., D. He. 2004. Solving the n-job 3-stage flexible flowshop scheduling problem using an agent-based approach. *International Journal of Production Research* **42**(4) 777-799.

Baker, A.D. 1998. A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: dispatching, scheduling, and pull. *Journal of Manufacturing Systems* **17**(4) 297-320.

Baker, C.T., B.P. Dzielinski. 1960. Simulation of a simplified job shop. *Management Science* **6** 311-323.

Blackstone, J.H., D.T. Phillips, G.L. Hogg. 1982. A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *International Journal of Production Research* **20**(1) 27-45.

Conway, R.W., B.M. Johnson, L.W. Maxwell. 1960. An experimental investigation of priority dispatching. *Journal of Industrial Engineering* **11** 221-230.

Conway, R.W. 1965. Priority dispatching and job lateness in a job shop. *Journal of Industrial Engineering* **16**(4) 228-237.

Conway, R.W., W.L. Maxwell, L.W. Miller. 1967. *Theory of Scheduling.* Addison-Wesley, Boston, Massachusetts, USA.

Dabbas, R.M., J.W. Fowler. 2003. A new scheduling approach using combined dispatching criteria in wafer fabs. *IEEE Transactions on Semiconductor Manufacturing* **16**(3) 501-510.

Davis, R., R. Smith. 1983. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence* **20** 63-109.

Dewan, P., S. Joshi. 2001. Implementation of an auction-based distributed scheduling model for a dynamic job shop environment. *International Journal of Computer Integrated Manufacturing* **14**(5) 446-456.

Dewan, P., S. Joshi. 2002. Auction-based distributed scheduling in a dynamic job shop environment. *International Journal of Production Research* **40**(5) 1173-1191.

Duffie, N.A. 1990. Synthesis of heterarchical manufacturing systems. *Computers in Industry* **14** 167-174.

Duffie, N.A., R. Chitturi, J. Mou. 1988. Fault-tolerant heterarchical control of heterogeneous manufacturing system entities. *Journal of Manufacturing Systems* **7** 315-327.

Duffie, N.A., R.S. Piper. 1986. Nonhierarchical control of manufacturing systems. *Journal of Manufacturing Systems* **5** 137-139.

Duffie, N.A., R.S. Piper. 1987. Nonhierarchical control of a flexible manufacturing cell. *Robotics and Computer-integrated Manufacturing Systems* **3** 175-179.

Duffie, N.A., V.V. Prabhu. 1994. Real-time distributed scheduling of heterarchical manufacturing systems. *Journal of Manufacturing Systems* **13** 94-107.

Haupt, R. 1989. A survey of priority rule-based scheduling. *OR Spektrum* **11** 3-16.

Holthaus, O., C. Rajendran. 1999. A comparative study of dispatching rules in dynamic flowshops and jobshops. *European Journal of Operations Research* **116** 156-170.

Holthaus, O., C. Rajendran. 2000. Efficient jobshop dispatching rules: further developments. *Production Planning and Control* **11**(2) 171-178.

Hung, Y.F., C.B. Chang. 1999. Using an empirical queuing approach to predict future flow times. *Computers and Industrial Engineering* **37**(4) 809-821.

Hung, Y.F., C.B. Chang. 2002. Dispatching rules using flow time predictions for semiconductor wafer fabrications. *Journal of Chinese Institute of Industrial Engineers* **19**(1) 67-74.

Hutchinson, J. 1991. Current issues concerning FMS scheduling. *OMEGA* **19** 529-537.

Ito, K. 1951. On stochastic differential equations. *Memoirs American Mathematical Society* **4** 1-51.

Jayamohan, M.S., C. Rajendran. 2000. New dispatching rules for shop scheduling: a step forward. *International Journal of Production Research* **38**(3) 563-586.

Kutanoglu, E., S.D. Wu. 1999. On combinatorial auction and Lagrangean relaxation for distributed resource scheduling. *IIE Transactions* **31** 813-826.

Law, A.M., W.D. Kelton. 1984. Confidence intervals for steady-state simulations: I. A survey of fixed sample size procedures. *Operations Research* **32** 1221-1239.

Lee, C.-Y., L. Lei, M. Pinedo. 1997. Current trends in deterministic scheduling. *Annals of Operations Research* **70**(0) 1-41.

Lee, H.T., S.H. Chen, H.Y. Kang. 2002. Multicriteria scheduling using fuzzy theory and tabu search. *International Journal of Production Research* **40**(5) 1221-1234.

Lin, C.Y. 1996. Shop floor scheduling of semiconductor wafer fabrication using real-time feedback control and prediction. Ph.D. Dissertation, Industrial Engineering and Operations Research, University of California-Berkeley, Berkeley, California, USA.

Lin, G., J. Solberg. 1992. Integrated shop floor control using autonomous agents. *IEEE Transactions* **24**(3) 57-71.

Lu, S.C.H., D. Ramaswamy, P.R. Kumar. 1994. Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants. *IEEE Transactions on Semiconductor Manufacturing* **7**(3) 374-388.

Malone, T.R., R.E. Fikes, M.T. Howard. 1983. Enterprise: a market-like task scheduler for distributed environments. Internal Report, Center for Information Research, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

Morton, T.E., D.W. Pentico. 1993. Heuristic scheduling systems: with applications to production systems and project management. Wiley, New York.

Nanot, Y.R. 1963. An experimental investigation and comparative evaluation of priority disciplines in job shop-like queuing networks. Research Report No. 87, Management Sciences Research Project, University of California-Los Angeles, Los Angeles, California, USA.

Oksendal, B. 1998. *Stochastic Differential Equations: An Introduction with Applications*, 5th ed. Springer-Verlag, Berlin/Heidelberg, Germany.

Panwalkar, S.S., W. Iskander. 1977. A survey of scheduling rules. *Operations Research* **25**(1) 45-61.

Pechoucek, M., A. Riha, J. Vokrinek, V. Marik, V. Prazma. 2002. ExPlanTech: applying multi-agent systems in production planning. *International Journal of Production Research* **40**(15) 3681-3692.

Pinedo, M. 1995. *Scheduling: Theory, Algorithms, and Systems.* Prentice-Hall, Englewood Cliffs, New Jersey, USA.

Raghu, T.S., C. Rajendran. 1993. An efficient dynamic dispatching rule for scheduling in a job shop. *International Journal of Production Economics* **32** 301-313.

Rai, R., S. Kameshwaran, M.K. Tiwari. 2002. Machine-tool selection and operation allocation in FMS: solving a fuzzy goal-programming model using a genetic algorithm. *International Journal of Production Research* **40**(3) 641-665.

Ramasesh, R. 1990. Dynamic job shop scheduling: a survey of simulation research. *OMEGA* **18** 43-57.

Rochette, R., R.P. Sadowski. 1976. A statistical comparison of the performance of simple dispatching rules for a particular set of jobs. *International Journal of Production Research* **14** 63-75.

Rossner, P.R. 2005. Entwurf globaler Prioritaetsregeln fuer dynamisch stochastische Werkstattfertigungssysteme. Diploma Thesis, Department of Production and Logistics, University of Passau, Passau, Germany.

Sabuncuoglu, I., D.L. Hommertzheim. 1992. Dynamic dispatching algorithm for scheduling machines and AGVs in a flexible manufacturing system. *International Journal of Production Research* **30** 1059-1080.

Sabuncuoglu, I., S. Karabuk. 1998. A beam search-based algorithm and evaluation of scheduling approaches for flexible manufacturing systems. *IIE Transactions* **30** 179-191.

Shaw, M. 1988. A distributed knowledge-based approach to flexible automation: the contract net framework. *International Journal of Flexible Manufacturing Systems* **1** 85-104.

Shaw, M., A. Whinston. 1985. Task bidding and distributed planning in flexible manufacturing. *IEEE Transactions* **2** 184-189.

Siwamogsatham, T., C. Saygin. 2004. Auction-based distributed scheduling and control scheme for flexible manufacturing systems. *International Journal of Production Research* **42**(3) 547-572.

Smith, R.J., R. Davis. 1981. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man and Cybernetics* **11**(1) 61-70.

Srinivas, M., K. Tiwari, V. Allada. 2004. Solving the machine-loading problem in a flexible manufacturing system using a combinatorial auction-based approach. *International Journal of Production Research* **42**(9) 1879-1893.

Stecke, K.E., J. Solberg. 1981. Loading and control policies for a flexible manufacturing system. *International Journal of Production Research* **19** 481-490.

Suresh, V., D. Chaudhuri. 1993. Dynamic scheduling: a survey of research. *International Journal of Production Economics* **32** 53-63.

Upton, D., M. Barash, M. Matheson. 1991. Architectures and auctions in manufacturing. *International Journal of Computer Integrated Manufacturing* **4**(1) 23-33.

Valkenaer, P., V.H. Brussel, L. Bongaerts, J. Wyns. 1994. Results of the holonic control system benchmark at the KU Leuven. Proceedings of the CIMAT Conference, Rensselaer Polytechnic Institute, Troy, New York, USA. 128-133.

Welch, P.D. 1981. On the problem of the initial transient in steady-state simulation. IBM Watson Research Center, Yorktown Heights, New York, USA.

Wu, S.D., E.-S. Byeon, R.H. Storer. 1999. A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. *Operations Research* **47**(1) 113-124.