FEP WORKING PAPERS
FEP WORKING PAPERS

Research Work in Progress

n. 234, Dec. 2006

Heuristics for the single machine scheduling problem with early and quadratic tardy penalties

Jorge M. S. Valente

LIACC/NIAAD, Faculdade de Economia

U.PORTO

FEP FACULDADE DE ECONOMIA UNIVERSIDADE DO PORTO

Heuristics for the single machine scheduling problem with early and quadratic tardy penalties

Jorge M. S. Valente*

LIACC/NIAAD

Faculdade de Economia, Universidade do Porto, Portugal

December 19, 2006

Abstract

In this paper, we consider the single machine scheduling problem with linear earliness and quadratic tardiness costs, and no machine idle time. We propose several dispatching heuristics, and analyse their performance on a wide range of instances. The heuristics include simple scheduling rules, as well as a procedure that takes advantage of the strengths of these rules. We also consider linear early / quadratic tardy dispatching rules, and a greedy-type procedure.

Extensive experiments were performed to determine appropriate values for the parameters required by some of the heuristics. The computational tests show that the best results are given by the linear early / quadratic tardy dispatching rule. This procedure is also quite efficient, and can quickly solve even very large instances.

Keywords: heuristics; scheduling; single machine; early penalties; quadratic tardy penalties; no machine idle time; dispatching rules.

*Address: Faculdade de Economia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-464 Porto, Portugal. Fax: + 351 22 550 50 50. E-mail: jvalente@fep.up.pt.

1 Introduction

In this paper, we consider a single machine scheduling problem with linear earliness and quadratic tardiness costs, and no machine idle time. Formally, the problem can be stated as follows. A set of n independent jobs $\{J_1, J_2, \cdots, J_n\}$ has to be scheduled on a single machine that can handle at most one job at a time. The machine is assumed to be continuously available from time zero onwards, and preemptions are not allowed. Job $J_j, j = 1, 2, \cdots, n$, requires a processing time p_j and should ideally be completed on its due date d_j. For a given schedule, the earliness and tardiness of J_j are respectively defined as $E_j = \max\{0, d_j - C_j\}$ and $T_j = \max\{0, C_j - d_j\}$, where C_j is the completion time of J_j. The objective is then to find a schedule that minimizes the sum of linear earliness and quadratic tardiness costs $\sum_{j=1}^{n} E_j + T_j^2$, subject to the constraint that no machine idle time is allowed.

Scheduling models with a single processor may appear to arise infrequently in practice. However, this scheduling environment does indeed occur in several activities (for a recent example in the chemical industry, see Wagner et al. (2002)). Moreover, the performance of many production systems is quite often dictated by the quality of the schedules for a single bottleneck machine. Models with a single processor are then most useful in practice for scheduling such a machine. Also, the analysis of single machine problems provides insights that prove valuable for scheduling more complex systems. In fact, multiple processor systems can sometimes be relaxed to a single machine problem, or a sequence of such problems. Furthermore, the solution procedures for some complex systems, such as job shop environments, often require solving single machine subproblems.

Scheduling models with both earliness and tardiness penalties are compatible with the philosophy of just-in-time (JIT) production. The JIT production philosophy emphasizes producing goods only when they are needed, and therefore takes up the view that both earliness and tardiness should be discouraged. For that reason, an ideal schedule is one in which all jobs are completed exactly on their due dates. Earliness/tardiness problems are also compatible with a recent trend in industry, namely the adoption of supply chain management by many organisations. This approach seeks to integrate the flow of materials from the suppliers to the customers, in order to improve the efficiency of the supply chain and to provide a better service to the end user. The adoption of this approach has caused organisations to view early deliveries, in addition to tardy deliveries, as undesirable.

In this paper, we consider linear earliness and quadratic tardiness costs. In fact, early deliveries or early completions of jobs result in unnecessary inventory that ties up cash, as well as space and resources required to maintain and manage the inventory. These costs tend in many cases to be linear with the quantity of inventory, hence the linear penalty for early jobs. On the other hand, late deliveries or jobs that are completed after their due dates result in lost sales and loss of goodwill, as well as disruptions and delays in stages further down the supply chain or production line. In many situations, these costs increase in severity in a quadratic fashion, leading to a quadratic penalty for tardy jobs.

We assume that no machine idle time is allowed. This assumption is actually appropriate for many production settings. Indeed, when the capacity of the machine is limited when compared with the demand, the machine must be kept running in order to satisfy the customers' orders. Idle time must also be avoided for machines with high operating costs, since the cost of keeping the machine running is then higher than the earliness cost incurred by completing a job before its due date. Furthermore, the assumption of no idle time is also justified when starting a new production run involves high setup costs or times. Some specific examples of production settings where the no idle time assumption is appropriate have been given by Korman (1994) and Landis (1993). More specifically, Korman considers the Pioneer Video Manufacturing (now Deluxe Video Services) disc factory at Carson, California, while Landis analyses the Westvaco envelope plant at Los Angeles.

The corresponding problem with inserted idle time has been previously considered by Schaller (2004). He presented a timetabling procedure to optimally insert idle time in a given sequence, as well as a branch-and-bound procedure and simple and efficient heuristic algorithms. The single machine early/tardy problem with linear earliness and tardiness costs $\sum_{j=1}^{n} E_j + T_j$

has also been previously considered by Garey et al. (1988) and Kim and Yano (1994). Garey et al. (1988) show that the problem is NP-hard, and propose a timetabling procedure. Kim and Yano (1994) present some properties of optimal solutions, and use them to develop both optimal and heuristic algorithms.

The minimization of the quadratic lateness $\sum_{j=1}^{n} L_j^2$, where the lateness of J_j is defined as $L_j = C_j - d_j$, has also been previously considered. Gupta and Sen (1983) presented a branch-and-bound algorithm and a heuristic rule for the problem with no idle time. Su and Chang (1998) and Schaller (2002) considered the insertion of idle time, and proposed timetabling procedures and heuristic algorithms. Sen et al. (1995) presented a branch-and-bound algorithm for the weighted problem $\sum_{j=1}^{n} w_j L_j^2$ where idle time is allowed only prior to the start of the first job. Baker and Scudder (1990) provide an excellent survey of scheduling problems with earliness and tardiness penalties, while Kanet and Sridharan (2000) give a review of scheduling models with inserted idle time that complements our focus on a problem with no machine idle time.

In this paper, we propose several dispatching heuristics, and analyse their performance on a large set of instances. We consider three simple but widely used scheduling rules, and propose an adaptation of one of those rules to a quadratic tardiness objective function. A heuristic that tries to take advantage of the strengths of the best-performing of these simple rules is also developed. We also present modified versions of early/tardy dispatching procedures originally proposed for the weighted problem with fully linear costs. These heuristics have been suitably adapted, in order to take into account the quadratic tardiness cost, as well as the non-weighted nature of the considered problem. Finally, we also present a greedy-type heuristic procedure. Extensive computational experiments were performed in order to determine appropriate values for the parameters required by some of the heuristics.

The remainder of this paper is organized as follows. The heuristics are described in section 2. In section 3, we present the computational results. Finally, some concluding remarks are provided in section 4.

2 The heuristics

2.1 Simple linear dispatching rules

We consider three simple scheduling rules, namely the longest processing time (LPT), earliest due date (EDD) and shortest processing time (SPT) heuristics. The LPT (SPT) rule schedules the jobs in non-increasing (non-decreasing) order of their processing times, while the EDD heuristic sequences the jobs in non-decreasing order of their due dates. These rules only require sorting, and their time complexity is then $O(n \log n)$.

These heuristics are included for two major reasons. On the one hand, these rules are quite well-known and widely used in many production settings. Therefore, it seems reasonable to include them for comparison purposes.

On the other hand, these rules have some interesting properties for the related problem with a fully linear objective function $\sum_{j=1}^{n} E_j + T_j$. Indeed, the LPT heuristic is particularly adequate to problems where most jobs will be completed early. In fact, the LPT sequence is optimal if it does not contain any tardy jobs. Conversely, the SPT rule is optimal if it generates a schedule with no early jobs. Therefore, this rule is appropriate for problems where most jobs will be tardy.

Finally, the EDD heuristic usually performs better than either the LPT or SPT rules when the number of early and tardy jobs is relatively balanced. Therefore, each one of these simple rules can perform quite well, under the appropriate circumstances, for the problem with a completely linear objective function. For this reason, it seems appropriate to analyse their performance for the problem with a quadratic tardiness cost.

2.2 Simple quadratic dispatching rule

The SPT rule is locally optimal, under the appropriate conditions, for the linear total tardiness problem $\sum_{j=1}^{n} T_j$. In fact, it two adjacent jobs are always tardy, regardless of their order, it is optimal to schedule those jobs in SPT order. In this section, we present a dispatching rule that is derived from a local optimality condition for the quadratic tardiness problem $\sum_{j=1}^{n} T_j^2$.

Therefore, this heuristic is an adaptation of the SPT rule to a quadratic objective function.

Theorem 1 *Consider any two adjacent jobs i and j that are always tardy, regardless of their order. In an optimal sequence, all such adjacent pairs of jobs must satisfy the following condition:*

$$(1/p_i)\left[p_j + 2\left(t + p_i - d_i\right)\right] \geq (1/p_j)\left[p_i + 2\left(t + p_j - d_j\right)\right],$$

where job i immediately precedes job j, and t is the start time of job i.

Proof. The condition can be established using simple interchange arguments. For the sake of brevity, we omit the details. ■

Theorem 1 provides a local optimality condition for two adjacent jobs that are always tardy, regardless of their order. The left (right) side of this expression can be interpreted as the priority of job i with respect to job j (job j with respect to job i) at time t. A dispatching rule priority index can then be derived by comparing the priority of each job with an average job with processing time $\overline{p}$, where $\overline{p}$ is the average processing time of the remaining unscheduled jobs. Therefore, the priority index of job j at time t, denoted as $I_j(t)$, can be calculated as:

$$I_j(t) = (1/p_j)\left[\overline{p} + 2\max\left(t + p_j - d_j, 0\right)\right].$$

At each iteration, the SPT_s_j dispatching rule selects the unscheduled job with the largest priority. The priority index of the SPT_s_j heuristic includes both a shortest processing time component (SPT) and a slack (s_j) related component (the slack of job j is defined as $s_j = d_j - t - p_j$). When a job is early, the SPT_s_j heuristic is equivalent to the SPT rule, since the priority of job j is then equal to $(1/p_j)\,\overline{p}$. When a job is tardy, however, the SPT ratio $(1/p_j)$ is modified by a slack-related component, and the priority increases with the job's tardiness.

The SPT_s_j dispatching heuristic is particularly suited to problems where most jobs will be completed after their due dates, since it is derived from a local optimality condition for tardy jobs. Therefore, the SPT_s_j rule is essentially an adaptation of the SPT heuristic to a quadratic tardiness objective. The time complexity of the SPT_s_j heuristic is $O(n^2)$.

2.3 The CS heuristic

We performed early computational tests with the LPT, EDD, SPT and SPT_s_j dispatching rules. These tests showed that the SPT_s_j heuristic indeed performed better than the SPT rule. Moreover, the preliminary tests also showed that the best results were given by the EDD (SPT_s_j) heuristic for problems where most jobs were early (tardy). The LPT heuristic was outperformed by the other procedures, even for instances where most jobs were early. In fact, the LPT heuristic focuses on minimizing the earliness costs, and completely disregards the tardiness component of the objective function. This means that the LPT sequence may contain a few jobs that are quite tardy, even for instances where most jobs will indeed be early. Since the objective function penalty for tardiness is much higher than the penalty for earliness, the LPT sequence will then have a large cost, even though it minimizes the earliness component.

In this section, we present a heuristic (denoted as CS) that tries to take advantage of the strengths of the EDD and SPT_s_j rules. At each iteration, the CS heuristic uses one of these two rules to choose the next job. The choice of dispatching rule is based on the characteristics of the current workload. Indeed, at each iteration the CS heuristic selects the rule that is expected to provide the best performance, given the characteristics of the current set of unscheduled jobs.

The CS procedure classifies the current workload as non-tardy or tardy. When most jobs have large slacks, the current workload is classified as non-tardy. Conversely, a tardy load consists mainly of jobs with low slacks. At each iteration, the CS heuristic analyses the characteristics of the current set of unscheduled jobs, and classifies the workload as either non-tardy or tardy. Then, the CS procedure selects the EDD (SPT_s_j) rule when the load is non-tardy (tardy).

We considered two versions of the CS heuristic. These versions share the same basic framework that was just described, and differ only in the criterion used to classify the workload as non-tardy or tardy. In both versions, we first calculate a critical slack value $crit\_slack$ (in fact, CS stands for Critical Slack). This critical value is calculated as $crit\_slack = slack\_prop * n_U * \overline{p}$, where n_U is the number of unscheduled jobs, and $0 \leq slack\_prop < 1$ is a user-defined parameter. Therefore, the critical slack value is then a proportion $slack\_prop$ of the total processing time of the currently unscheduled jobs.

The CS_AS version calculates the average slack $\overline{s}$ of the remaining unscheduled jobs. The workload is then classified as non-tardy (tardy) if $\overline{s} > crit\_slack$ ($\overline{s} \leq crit\_slack$). In the CS_LP version, on the other hand, each job is first classified as non-tardy or tardy. A job is said to be non-tardy (tardy) if $s_j > crit\_slack$ ($s_j \leq crit\_slack$). The proportion of non-tardy and tardy jobs is then calculated, and the current workload is classified as non-tardy (tardy) if the percentage of non-tardy (tardy) jobs is the largest. The time complexity of both versions of the CS heuristic is $O(n^2)$.

2.4 Linear early / quadratic tardy dispatching rules

Ow and Morton (1989) developed two early/tardy dispatching rules, denoted as LINET and EXPET, for the fully linear problem with job-dependent earliness and tardiness penalties $\sum_{j=1}^{n} h_j E_j + w_j T_j$ (where h_j and w_j are the job-specific earliness and tardiness penalties, respectively). In this section, we propose adaptations of these rules to the linear earliness and quadratic tardiness problem. Therefore, we suitably modified the heuristics proposed by Ow and Morton in order to take into account the quadratic tardiness cost, as well as the fact that $h_j = w_j = 1$. The proposed heuristics are denoted by EQTP_LIN and EQTP_EXP, where EQTP stands for Earliness and Quadratic Tardiness Penalties.

Both versions of the EQTP dispatching rule calculate a priority index for each remaining job every time the machine becomes available, and the job with the highest priority is then selected to be processed next. Let $I_j(t)$

denote the priority index of job J_j at time t. The EQTP_LIN version then uses the following priority index $I_j(t)$:

$$I_j(t) = \begin{cases} (1/p_j)\left[\overline{p} + 2(t + p_j - d_j)\right] & \text{if} \quad s_j \leq 0 \\ (\overline{p}/p_j) - (1/p_j)(\overline{p}+1)\, s_j/k\overline{p} & \text{if} \quad 0 < s_j < k\overline{p} \\ -(1/p_j) & \text{otherwise,} \end{cases}$$

where k is a lookahead parameter and s_j and $\overline{p}$ are as previously defined.

The EQTP_EXP rule instead uses the following priority index:

$$I_j(t) = \begin{cases} (1/p_j)\left[\overline{p} + 2(t + p_j - d_j)\right] & \text{if} \quad s_j \leq 0 \\ (\overline{p}/p_j)\exp\left[-(\overline{p}+1)\, s_j/k\overline{p}\right] & \text{if} \quad 0 < s_j < \left[\overline{p}/(\overline{p}+1)\right] k\overline{p} \\ (1/p_j)^{-2}\left[(\overline{p}/p_j) - (1/p_j)(\overline{p}+1)\, s_j/k\overline{p}\right]^3 & \text{if} \quad \left[\overline{p}/(\overline{p}+1)\right] k\overline{p} \leq s_j < k\overline{p} \\ -(1/p_j) & \text{otherwise,} \end{cases}$$

where s_j, $\overline{p}$ and k are as previously defined.

The EQTP_LIN and EQTP_EXP dispatching rules assign a priority value of $-(1/p_j)$ to jobs that are in no danger of becoming tardy ($s_j \geq k\overline{p}$). This assures that two jobs that have large slacks will be scheduled in LPT order. Conversely, the SPT_s_j rule is used to calculate the priority value when a job is on time or late ($s_j \leq 0$). The EQTP_LIN and EQTP_EXP heuristics differ in the calculation of the job priorities for the intermediate values of the job slack. The priority decreases linearly as the job slack increases in the EQTP_LIN dispatching rule, while exponential and cubic functions are instead used in the EQTP_EXP heuristic.

The effectiveness of the EQTP_LIN and EQTP_EXP heuristics depends on the value of the lookahead parameter k. This parameter should reflect the number of competing critical jobs, i.e., the number of jobs that may clash each time a sequencing decision is to be made (for details, see Ow and Morton (1989)). In the proposed implementation, the value of k is calculated dynamically at each iteration. Therefore, each time a scheduling decision has to be made, the characteristics of the current workload are used to determine an appropriate value for the lookahead parameter.

The following procedure is then used to calculate the value of the lookahead parameter k at each iteration. First, a critical slack value $crit\_slack$ is calculated, just as previously described for the CS heuristics. Then, each job is classified as critical if $0 < s_j \leq crit\_slack$, and non-critical otherwise. Therefore, a job is considered critical if it is not already tardy ($s_j > 0$), but is about to become tardy ($s_j \leq crit\_slack$). Finally, the lookahead parameter k is set equal to the number of critical jobs. The time complexity of the EQTP_LIN and EQTP_EXP dispatching rules is $O(n^2)$.

2.5 Greedy heuristic

In this section, we present a greedy-type procedure, denoted by Greedy. This heuristic is an adaptation of a procedure originally introduced by Fadlalla et al. (1994) for the mean tardiness problem, and later adapted to other problems (see, for instance, Volgenant and Teerhuis (1999) and Valente and Alves (2005)).

We considered two different versions of the Greedy heuristic. These versions share the basic framework, and differ only slightly in the calculation of the job priorities. Let c_{xy}, with $x \neq y$, be the combined cost of scheduling jobs J_x and J_y, in this order, in the next two positions in the sequence, i.e., c_{xy} is the sum of the costs of J_x and J_y when they are completed at times $t + p_x$ and $t + p_x + p_y$, respectively. Also, let L be a list with the indexes of the yet unscheduled jobs and $P(j)$ the priority of job J_j. The steps of the Greedy_v1 version are:

Step 1. Initialization:

Set $t = 0$ and $L = \{1, 2, \ldots, n\}$.

Step 2. Calculate the job priorities:

Set $P(j) = 0$, for all $j \in L$;

For all pairs of jobs $(i, j) \in L$, with $i < j$, do:

Calculate c_{ij} and c_{ji};

If $c_{ij} < c_{ji}$, set $P(i) = P(i) + 1$;

If $c_{ij} > c_{ji}$, set $P(j) = P(j) + 1$;

If $c_{ij} = c_{ji}$, set $P(i) = P(i) + 1$ and $P(j) = P(j) + 1$.

Step 3. Select the next job:

Schedule job l for which $P(l) = \max\{P_j; j \in L\}$;

Set $t = t + p_l$ and $L = L \setminus \{l\}$.

Step 4. Stopping condition:

If $|L| = 1$, stop;

Else, go to step 2.

In the Greedy_v2 version, Step 2 is instead given by:

Step 2. Calculate the job priorities:

Set $P(j) = 0$, for all $j \in L$;

For all pairs of jobs $(i, j) \in L$, with $i < j$, do:

Calculate c_{ij}, c_{ji} and $|c_{ij} - c_{ji}|$;

If $c_{ij} < c_{ji}$

set $P(i) = P(i) + |c_{ij} - c_{ji}|$;

set $P(j) = P(j) - |c_{ij} - c_{ji}|$.

Else

set $P(i) = P(i) - |c_{ij} - c_{ji}|$;

set $P(j) = P(j) + |c_{ij} - c_{ji}|$.

If $c_{ij} < c_{ji}$, it seems better to schedule job J_i in the next position rather than job J_j. Conversely, it seems preferable to schedule job J_j next when $c_{ij} > c_{ji}$. In the Greedy_v1 version, the priority $P(j)$ of job J_j is therefore the number of times job J_j is the preferred job for the next position when it is compared with all the other unscheduled jobs. In the Greedy_v2 version, for all pairs of jobs (i, j), with $i < j$, the priority of the preferred job is instead increased by $|c_{ij} - c_{ji}|$, while the priority of the other job is decreased by that same value. The time complexity of both versions of the Greedy heuristic is $O(n^3)$.

3 Computational results

In this section, we first present the set of test problems used in the computational tests, and then describe the preliminary computational experiments. These experiments were performed to determine appropriate values for the parameters required by the CS and EQTP heuristics. Moreover, the performance of the alternative versions of the CS, EQTP and Greedy heuristics was also analysed in these initial experiments, in order to select the best-performing of those versions. Finally, we present the computational results. We first compare the heuristic procedures, and then evaluate the heuristic results against optimum objective function values for some instance sizes. Throughout this section, and in order to avoid excessively large tables, we will sometimes present results only for some representative cases.

3.1 Experimental design

The computational tests were performed on a set of problems with 10, 15, 20, 25, 30, 40, 50, 75, 100, 250, 500, 750, 1000, 1500 and 2000 jobs. These problems were randomly generated as follows. For each job J_j, an integer processing time p_j was generated from one of the two uniform distributions $[1, 10]$ and $[1, 100]$, in order to obtain a low (L) and a high (H) range, respectively, for the processing time values. For each job J_j, an integer due date d_j is generated from the uniform distribution $[P(1 - T - R/2), P(1 - T + R/2)]$, where P is the sum of the processing times of all jobs, T is the tardiness factor, set at 0.0, 0.2, 0.4, 0.6, 0.8 and 1.0, and R is the range of due dates, set at 0.2, 0.4, 0.6 and 0.8.

For each combination of problem size n, processing time range (rng), T and R, 50 instances were randomly generated. Therefore, a total of 1200 instances were generated for each combination of problem size and processing time range. All the algorithms were coded in Visual C++ 6.0, and executed on a Pentium IV - 2.8 GHz personal computer. Due to the large computational times that would be required, the Greedy heuristic was only applied to instances with up to 500 jobs.

3.2 Parameter adjustment tests

In this section, we describe the preliminary computational experiments. These initial experiments were performed to determine appropriate values for the parameters required by the CS_AS, CS_LP, EQTP_LIN and EQTP_EXP dispatching rules. The performance of the alternative versions of the CS, EQTP and Greedy heuristics was also analysed, in order to select the best-performing versions. A separate problem set was used to conduct these preliminary experiments. This test set included instances with 25, 50, 100, 250, 500, 1000 and 2000 jobs, and contained 5 instances for each combination of instance size, processing time range, T and R. The instances in this smaller test set were generated randomly just as previously described for the full problem set.

Extensive computational tests were performed to determine an appropriate value for the *slack_prop* parameter used by the CS_AS, CS_LP, EQTP_LIN and EQTP_EXP heuristics. We considered the values $\{0.00, 0.05, 0.10, \ldots, 0.95\}$, and computed the objective function value for each *slack_prop* value and each instance. An analysis of these results showed that a value of $slack\_prop = 0.20$ provided the best performance for the CS_AS and CS_LP heuristics. For the EQTP_LIN and EQTP_EXP dispatching rules, the best results were given by *slack_prop* values in the range $[0.50, 0.95]$. We then decided to set *slack_prop* at 0.55 and 0.60 for the EQTP_LIN and the EQTP_EXP heuristics, respectively, since these values consistently provided good results for all instance types.

The performance of the alternative versions of the CS, EQTP and Greedy heuristics was also analysed in these preliminary computational experiments, in order to select the best-performing versions. The CS_AS version was then evaluated against its CS_LP counterpart. Also, the EQTP_EXP version was compared with its EQTP_LIN alternative. Finally, we compared the Greedy_v1 and Greedy_v2 versions. Therefore, we compared the following three ($h1$ vs $h2$) pairs of alternative heuristic versions: (CS_AS vs CS_LP), (EQTP_EXP vs EQTP_LIN) and (Greedy_v1 vs Greedy_v2).

In table 1, we present the average of the relative improvements in objective function value provided by the $h1$ heuristic over its $h2$ counterpart (%imp), as well as the percentage number of times version $h1$ performs better ($<$), equal ($=$) or worse ($>$) than version $h2$. The relative improvement given by version $h1$ is calculated as $(h2\_ofv - h1\_ofv)/h2\_ofv \times 100$, where $h2\_ofv$ and $h1\_ofv$ are the objective function values of the appropriate heuristic versions.

The performance of the alternative versions of the CS heuristic is quite similar. In fact, the objective function values provided by these alternative versions is generally quite close, particularly for the medium and large size instances. The CS_AS version, however, usually provides better results than its CS_LP counterpart for a slightly larger number of instances.

The EQTP_EXP heuristic performs better than its EQTP_LIN alternative. Indeed, the EQTP_EXP version provides on average a relative improvement in the objective function value of over 1% (3%) for instances with a low (high) processing time range. Also, the EQTP_EXP version gives better results for a larger number of the test instances.

The Greedy_v1 version clearly outperforms its Greedy_v2 alternative. In fact, the Greedy_v1 heuristic provides a relative improvement in the objective function value of about 4%-5% (with the exception of the instances with 25 jobs and a high range). Also, the Greedy_v1 version gives better results for over 80% (and in some cases, actually all) of the test instances. In the following sections, we will only present results for the CS_AS, EQTP_EXP and Greedy_v1 versions.

3.3 Heuristic results

In this section, we present the computational results for the heuristic procedures. In table 2, we give the average objective function value (ofv) for each heuristic, as well as the percentage number of times a heuristic provides the best result when compared with the other heuristics (%best). The average objective function values are calculated relative to the EQTP_EXP heuristic, and are therefore presented as index numbers. More precisely, these values are calculated as heur _ofv / eqtp_exp_ofv * 100, where heur_ofv and eqtp_exp_ofv are the average objective function values of the appropriate heuristic and the EQTP_EXP dispatching rule, respectively.

The best results are given by the EQTP_EXP dispatching rule, closely followed by the CS_AS procedure. In fact, the EQTP_EXP heuristic not only provides the lowest average objective function value, but also obtains the best results for a large percentage of the instances (particularly for the largest instances, or when the range of processing times is high). The CS_AS procedure also performs quite well, providing an average objective function value that is quite close to the results given by the EQTP_EXP heuristic.

The SPT_s_j and the Greedy_v1 heuristics also provide an adequate performance. The Greedy_v1 procedure gives an average objective function value that is about 2-3% worse than the EQTP_EXP heuristic, but it nevertheless provides the best results for a large number of instances. The SPT_s_j procedure performs well for medium and large instances, since it provides an average objective function value that is about 1% worse than the results given by the EQTP_EXP dispatching rule.

The simple LPT, EDD and SPT rules perform rather poorly, giving results that are substantially worse than those of the other heuristics. The linear SPT rule was clearly outperformed by its SPT_s_j quadratic counterpart. Hence, the modifications that were introduced in this linear rule, in order to adapt it to a quadratic objective function, have indeed significantly improved its performance. Therefore, it is certainly important to specifically address the quadratic tardiness component of the cost function, and develop a specific procedure, instead of simply using a heuristic appropriate for a linear tardiness objective function. The CS_AS heuristic performs considerably better than either of the EDD and SPT_s_j rules. Consequently, a considerable performance improvement can indeed be achieved by selectively using these two simple heuristics (i.e., by choosing at each iteration the rule that is expected to perform better, given the characteristics of the current job load).

In table 3, we present the effect of the T and R parameters on the average objective function value (once more calculated relative to the EQTP_EXP heuristic). This table gives results for the best heuristics (the EQTP_EXP is omitted, since its values would all be equal to 100) and for instances with 100 jobs. The SPT_s_j heuristic provides an average function value that is quite close to the results given by the EQTP_EXP dispatching rule for instances with a large tardiness factor T. The SPT_s_j rule, however, performs considerably worse when the tardiness factor is low. This result is to be expected, since the SPT_s_j heuristic is particularly suited to problems where most jobs will be completed after their due dates, since it is derived from a local optimality condition for tardy jobs. When the tardiness factor T is high, most jobs will be tardy, and the SPT_s_j rule indeed performs well. For low values of T, on the other hand, the proportion of tardy jobs is lower, and the performance of the SPT_s_j heuristic correspondingly deteriorates.

The CS_AS heuristic is quite close to the EQTP_EXP dispatching rule, and indeed sometimes even better, for $T \geq 0.4$. However, the CS_AS heuristic is clearly outperformed when most jobs are early. In fact, the CS_AS heuristic provides an average objective function value that is about 30% worse than the results given by the EQTP_EXP procedure when $T = 0.0$ or $T = 0.2$ and $R \geq 0.4$. The Greedy_v1 heuristic performs well for instances where most jobs are early ($T = 0.0$) or tardy ($T = 1.0$). The performance of the Greedy_v1 procedure then deteriorates as the tardiness factor T approaches its intermediate values. Therefore, the Greedy_v1 procedure is less effective when there is a greater balance between the number of early and tardy jobs.

In table 4, we present the heuristic runtimes (in seconds). The Greedy_v1 heuristic is computationally demanding, and therefore can only be used for small and medium size instances. The other heuristic procedures are quite fast, even for the largest instances. The simple LPT, EDD and SPT rules are the most efficient, since they only require sorting, which can be performed in $O(n \log n)$ time. The SPT_s_j and CS_AS procedures are also quite efficient, even with their higher $O(n^2)$ time complexity. The EQTP_EXP dispatching rule, even though it also requires $O(n^2)$ time, is more computationally demanding. Nevertheless, this heuristic is still extremely fast, being capable of solving even quite large instances with 2000 jobs in less than 0.3 seconds on a personal computer. The EQTP_EXP is then the heuristic procedure of choice, since it not only provides the best results, but is also computationally quite efficient.

3.4 Comparison with optimum results

In this section, we compare the heuristic results with the optimum objective function values for instances with up to 20 jobs. In table 5, we present the average of the relative deviations from the optimum (%dev), calculated as $(H - O)/O * 100$, where H and O are the heuristic and the optimum objective function values, respectively. The percentage number of times each heuristic generates an optimum schedule (%opt) is also given.

From table 5, we can see that the heuristics are much closer to the optimum for instances with a low processing time range, with the exception of the CS_AS procedure. The EQTP_EXP heuristic provides an acceptable performance for instances with a low processing time range, giving results that are 6-7% above the optimum. For instances with a high range, however, the deviation from the optimum ranges from 10% to 20% (though it decreases as the instance size increases).

The CS_AS heuristic gives results that are around 15% above the optimum. For the Greedy_v1 heuristic, the average deviation from the optimum is large for instances with a high processing time range. However, this heuristic provides an optimum solution for a large number of instances. In fact, for some problem sizes the Greedy_v1 procedure generates an optimum solution for over 50% of the instances.

In table 6, we present the effect of the T and R parameters on the relative deviation from the optimum. This table gives results for the best heuristics and for instances with 20 jobs. The SPT_s_j heuristic is quite close to the optimum for instances with a large tardiness factor T. The average relative deviation from the optimum, however, is substantially higher when the tardiness factor is low. This is to be expected, since the SPT_s_j heuristic is particularly suited to problems where most jobs will be tardy. When the tardiness factor T is high, most jobs will indeed be tardy, and the SPT_s_j rule is then quite close to optimal. For low values of T, on the other hand, the number of tardy jobs is lower, and the average deviation of the SPT_s_j heuristic from the optimum correspondingly increases.

The CS_AS dispatching rule is quite close to the optimum when the tardiness factor is greater than or equal to 0.6. The performance of the CS_AS heuristic, however, is clearly inferior when most jobs are early. In fact, the CS_AS heuristic is about 30-50% above the optimum for instances with $T = 0.0$ or $T = 0.2$. The effect of the T and R parameters on the average relative deviation from the optimum is similar for the EQTP_EXP and the Greedy_v1 heuristics. These procedures are much closer to the optimum when nearly all jobs are early ($T = 0.0$) or when there is a larger proportion of tardy jobs ($T \geq 0.6$). The performance of these dispatching rules deteriorates, particularly for the Greedy_v1 heuristic, when $T = 0.2$ or $T = 0.4$.

4 Conclusion

In this paper, we considered the single machine scheduling problem with linear earliness and quadratic tardiness costs, and no machine idle time. We proposed several dispatching heuristics, and analysed their performance on a wide range of instances. The heuristics included simple scheduling rules, as well as a procedure that takes advantage of the strengths of these rules. We also considered linear early / quadratic tardy dispatching rules, and a greedy-type procedure. Extensive computational experiments were performed to determine adequate values for the parameters required by some of the heuristics.

Dispatching heuristics are widely used in practice and, in fact, most real scheduling systems are either based on dispatching rules, or at least use them to some degree. Also, dispatching rules are often the only heuristic approach capable of generating solutions, within reasonable computation times, for large instances. Additionally, dispatching rules are used by other heuristic procedures, e.g., they are often used to generate the initial sequence required by local search or metaheuristic algorithms.

The best results were given by the EQTP_EXP dispatching rule. This heuristic provided the lowest average objective function values, and also obtained the best results for a large percentage of the instances. The performance of the EQTP_EXP procedure was most adequate for instances with a low processing time range, since it provided results that are about 6-7% above the optimum. For instances with a high range, however, the deviation from the optimum exceeded 10%, though it decreased as the instance size increased.

The Greedy_v1 heuristic is computationally demanding, and therefore can only be used for small and medium size instances. The other heuristic procedures, however, were quite fast, and are capable of solving even very large instances in less than one second on a personal computer. The EQTP_EXP dispatching rule is then the heuristic procedure of choice, since it not only provided the best results, but is also computationally quite efficient.

References

Baker, K. R. and G. D. Scudder (1990): Sequencing With Earliness and Tardiness Penalties: A Review. *Operations Research*, 38:22–36.

Fadlalla, A.; J. R. Evans; and M. S. Levy (1994): A Greedy Heuristic for the

Mean Tardiness Sequencing Problem. *Computers & Operations Research*, 21:329–336.

Garey, M. R.; R. E. Tarjan; and G. T. Wilfong (1988): One-Processor Scheduling with Symmetric Earliness and Tardiness Penalties. *Mathematics of Operations Research*, 13:330–348.

Gupta, S. K. and T. Sen (1983): Minimizing a Quadratic Function of Job Lateness on a Single Machine. *Engineering Costs and Production Economics*, 7:187–194.

Kanet, J. J. and V. Sridharan (2000): Scheduling with Inserted Idle Time: Problem Taxonomy and Literature Review. *Operations Research*, 48:99–110.

Kim, Y. D. and C. A. Yano (1994): Minimizing Mean Tardiness and Earliness in Single-Machine Scheduling Problems with Unequal Due Dates. *Naval Research Logistics*, 41:913–933.

Korman, K. (February 1994): A Pressing Matter. *Video*, pp. 46–50.

Landis, K. (1993): *Group Technology and Cellular Manufacturing in the Westvaco Los Angeles VH Department.* Project report in IOM 581, School of Business, University of Southern California.

Ow, P. S. and T. E. Morton (1989): The Single Machine Early/Tardy Problem. *Management Science*, 35:177–191.

Schaller, J. (2002): Minimizing the Sum of Squares Lateness on a Single Machine. *European Journal of Operational Research*, 143:64–79.

Schaller, J. (2004): Single Machine Scheduling with Early and Quadratic Tardy Penalties. *Computers & Industrial Engineering*, 46:511–532.

Sen, T.; P. Dileepan; and M. R. Lind (1995): Minimizing a Weighted Quadratic Function of Job Lateness in the Single Machine System. *International Journal of Production Economics*, 42:237–243.

Su, L.-H. and P.-C. Chang (1998): A Heuristic to Minimize a Quadratic Function of Job Lateness on a Single Machine. *International Journal of Production Economics*, 55:169–175.

Valente, J. M. S. and R. A. F. S. Alves (2005): Improved Heuristics for the Early/Tardy Scheduling Problem with No Idle Time. *Computers & Operations Research*, 32:557–569.

Volgenant, A. and E. Teerhuis (1999): Improved Heuristics for the n-job Single-Machine Weighted Tardiness Problem. *Computers & Operations Research*, 26:35–44.

Wagner, B. J.; D. J. Davis; and H. Kher (2002): The Production of Several Items in a Single Facility with Linearly Changing Demand Rates. *Decision Sciences*, 33:317–346.

	n	low rng				high rng			
	n	%imp	<	=	>	%imp	<	=	>
CS_AS	25	0.50	13.33	85.00	1.67	0.96	17.50	80.83	1.67
vs	50	0.07	14.17	80.83	5.00	0.25	20.00	73.33	6.67
CS_LP	100	0.03	15.00	73.33	11.67	0.07	21.67	66.67	11.67
	250	0.04	15.00	66.67	18.33	0.13	18.33	69.17	12.50
	500	0.01	17.50	61.67	20.83	0.03	20.00	61.67	18.33
	1000	0.01	20.83	60.00	19.17	0.03	24.17	65.00	10.83
	2000	0.01	23.33	60.83	15.83	-0.01	18.33	57.50	24.17
EQTP_EXP	25	1.43	44.17	32.50	23.33	3.29	60.00	20.00	20.00
vs	50	1.50	43.33	25.83	30.83	4.26	55.83	16.67	27.50
EQTP_LIN	100	1.35	43.33	25.00	31.67	3.88	52.50	20.00	27.50
	250	1.28	46.67	25.00	28.33	3.05	47.50	23.33	29.17
	500	1.25	43.33	26.67	30.00	3.25	46.67	25.00	28.33
	1000	1.73	44.17	25.83	30.00	2.87	45.83	25.00	29.17
	2000	1.76	41.67	29.17	29.17	2.72	45.83	25.00	29.17
Greedy_v1	25	3.57	82.50	14.17	3.33	-0.50	80.00	9.17	10.83
vs	50	4.47	91.67	7.50	0.83	4.19	85.00	0.83	14.17
Greedy_v2	100	4.85	97.50	1.67	0.83	5.70	83.33	0.00	16.67
	250	4.19	100.00	0.00	0.00	4.51	82.50	0.00	17.50
	500	3.97	100.00	0.00	0.00	5.51	82.50	0.00	17.50

Table 1: Heuristic version comparison

rng	heur	$n = 25$ ofv	$n = 25$ %best	$n = 100$ ofv	$n = 100$ %best	$n = 500$ ofv	$n = 500$ %best	$n = 2000$ ofv	$n = 2000$ %best
L	LPT	290.91	0.83	330.92	0.00	339.22	0.00	340.67	0.00
	EDD	130.40	2.33	135.20	2.00	136.20	1.33	136.68	1.92
	SPT	126.13	0.00	127.99	0.00	128.18	0.00	128.34	0.00
	SPT_s_j	103.09	6.17	101.50	1.83	100.93	11.33	100.83	28.42
	CS_AS	100.49	18.75	100.20	22.00	100.10	29.33	100.09	40.42
	EQTP_EXP	100.00	29.92	100.00	20.17	100.00	30.42	100.00	80.83
	Greedy_v1	101.58	65.42	102.68	59.00	102.86	61.00	—	—
H	LPT	325.02	0.08	364.43	0.00	377.41	0.00	380.17	0.00
	EDD	134.93	4.75	139.77	0.75	141.89	1.75	142.08	1.92
	SPT	129.45	0.00	133.08	0.00	134.78	0.00	134.82	0.00
	SPT_s_j	102.36	5.42	101.39	1.17	101.08	12.75	100.99	25.00
	CS_AS	100.19	19.42	100.18	9.00	100.14	18.08	100.13	24.50
	EQTP_EXP	100.00	44.58	100.00	56.75	100.00	54.58	100.00	91.83
	Greedy_v1	102.44	45.67	103.63	35.67	103.92	49.92	—	—

Table 2: Heuristic results

heur	T	low rng R=0.2	R=0.4	R=0.6	R=0.8	high rng R=0.2	R=0.4	R=0.6	R=0.8
SPT_s_j	0.0	186.39	182.35	180.59	176.31	204.42	202.17	215.09	207.42
	0.2	155.32	239.54	234.89	223.20	131.46	794.73	565.84	467.52
	0.4	115.73	155.28	242.17	551.25	119.35	156.01	280.71	1876.76
	0.6	103.97	110.23	113.81	108.60	104.68	112.71	115.49	109.19
	0.8	100.49	100.47	100.00	100.00	100.53	100.49	100.05	100.02
	1.0	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
CS_AS	0.0	132.33	130.22	126.91	124.17	135.53	133.37	131.02	126.46
	0.2	95.40	137.27	135.88	128.98	62.13	126.37	135.65	130.47
	0.4	99.30	99.19	97.99	91.68	100.37	101.10	101.79	109.29
	0.6	100.27	100.28	99.96	103.32	100.61	100.70	100.57	104.79
	0.8	100.49	100.42	100.04	100.15	100.52	100.44	100.08	100.31
	1.0	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Greedy_v1	0.0	99.86	99.57	99.33	98.77	100.35	100.98	101.82	103.93
	0.2	164.75	116.67	99.54	91.65	242.03	487.37	254.17	169.94
	0.4	172.15	190.76	224.77	260.04	183.55	217.47	337.38	1351.64
	0.6	127.69	124.38	107.36	100.87	132.66	132.32	125.99	101.88
	0.8	103.47	99.96	99.99	99.99	105.71	99.98	99.99	99.99
	1.0	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Table 3: Objective function value, relative to the EQTP_EXP heuristic, for instances with 100 jobs

var	heur	$n = 100$	$n = 250$	$n = 500$	$n = 1000$	$n = 1500$	$n = 2000$
L	LPT	0.0000	0.0001	0.0002	0.0003	0.0003	0.0007
	EDD	0.0000	0.0001	0.0001	0.0003	0.0003	0.0005
	SPT	0.0000	0.0001	0.0001	0.0003	0.0004	0.0005
	SPT_s_j	0.0001	0.0007	0.0031	0.0125	0.0278	0.0489
	CS_AS	0.0002	0.0010	0.0035	0.0138	0.0316	0.0569
	EQTP_EXP	0.0006	0.0044	0.0168	0.0683	0.1540	0.2728
	Greedy_v1	0.1840	2.8697	22.9598	—	—	—
H	LPT	0.0000	0.0001	0.0002	0.0004	0.0005	0.0007
	EDD	0.0000	0.0001	0.0001	0.0003	0.0003	0.0005
	SPT	0.0000	0.0001	0.0003	0.0002	0.0005	0.0005
	SPT_s_j	0.0001	0.0009	0.0029	0.0117	0.0262	0.0466
	CS_AS	0.0001	0.0009	0.0037	0.0150	0.0324	0.0584
	EQTP_EXP	0.0007	0.0042	0.0161	0.0635	0.1430	0.2563
	Greedy_v1	0.1848	2.8591	22.8678	—	—	—

Table 4: Heuristic runtimes (in seconds)

		$n = 10$		$n = 15$		$n = 20$	
rng	heur	%dev	%opt	%dev	%opt	%dev	%opt
L	LPT	437.16	3.92	616.09	2.33	854.98	1.08
	EDD	30.31	0.75	33.32	0.00	35.65	0.00
	SPT	168.93	0.17	215.78	0.17	275.99	0.00
	SPT_s_j	75.78	11.83	80.94	8.08	83.63	6.17
	CS_AS	15.85	12.58	16.09	8.67	16.75	6.08
	EQTP_EXP	6.93	27.58	6.65	16.75	7.24	11.42
	Greedy_v1	11.02	66.25	14.35	52.17	19.80	42.75
H	LPT	1659.80	2.17	2786.84	0.50	3755.46	0.50
	EDD	32.07	0.33	36.33	0.00	37.32	0.00
	SPT	589.29	0.00	810.08	0.00	1051.87	0.00
	SPT_s_j	195.66	7.17	230.05	5.25	224.85	3.50
	CS_AS	13.83	8.17	14.84	5.50	15.23	3.50
	EQTP_EXP	22.14	22.25	16.45	11.92	11.96	8.67
	Greedy_v1	40.18	52.67	68.90	35.83	92.85	30.33

Table 5: Comparison with optimum objective function values

heur	T	low rng R=0.2	R=0.4	R=0.6	R=0.8	high rng R=0.2	R=0.4	R=0.6	R=0.8
SPT_s_j	0.0	89.26	90.66	90.60	86.66	131.89	141.19	191.15	194.88
	0.2	124.43	205.61	255.51	202.92	125.65	620.08	1546.85	729.72
	0.4	31.22	73.04	205.97	477.64	22.86	64.63	185.59	1377.21
	0.6	7.75	14.99	19.19	27.09	5.60	12.68	17.66	22.90
	0.8	1.37	1.38	0.64	0.92	1.45	1.81	1.10	1.08
	1.0	0.04	0.07	0.03	0.06	0.05	0.07	0.09	0.11
CS_AS	0.0	33.70	32.54	31.93	26.19	37.15	33.52	36.63	31.75
	0.2	34.78	53.73	46.05	45.52	14.16	54.80	53.44	46.79
	0.4	7.72	10.32	20.07	39.68	3.43	4.52	9.36	21.71
	0.6	3.15	3.55	3.43	5.37	2.28	2.90	2.64	4.89
	0.8	1.37	1.32	0.62	0.78	1.45	1.80	0.98	0.91
	1.0	0.04	0.07	0.03	0.06	0.05	0.07	0.09	0.11
EQTP_EXP	0.0	0.20	0.58	1.31	2.10	0.66	1.64	2.65	2.64
	0.2	22.98	19.96	15.71	18.57	60.07	91.36	26.80	20.54
	0.4	10.79	11.65	22.34	36.80	6.34	4.57	13.49	44.53
	0.6	2.60	2.20	1.90	2.02	3.97	1.38	1.23	1.88
	0.8	1.10	0.45	0.15	0.25	1.68	0.82	0.30	0.25
	1.0	0.03	0.07	0.03	0.06	0.03	0.05	0.05	0.06
Greedy_v1	0.0	0.10	0.32	0.88	1.65	3.33	8.85	16.18	19.08
	0.2	46.49	24.62	13.41	25.40	127.37	265.48	287.64	371.61
	0.4	43.15	81.76	86.46	108.19	60.78	70.64	139.77	814.20
	0.6	15.09	11.31	6.45	8.89	18.66	17.10	5.51	1.05
	0.8	0.88	0.07	0.02	0.02	0.96	0.13	0.06	0.04
	1.0	0.01	0.02	0.00	0.01	0.00	0.00	0.02	0.01

Table 6: Relative deviation from the optimum for instances with 20 jobs

Recent FEP Working Papers

Nº 233	Pedro Cosme Vieira and Aurora A. C. Teixeira, *"Are Finance, Management, and Marketing Autonomous Fields of Scientific Research? An Analysis Based on Journal Citations"*, December 2006
Nº 232	Ester Gomes da Silva and Aurora A. C. Teixeira, *"Surveying structural change: seminal contributions and a bibliometric account"*, November 2006
Nº 231	Carlos Alves and Cristina Barbot, *"Do low cost carriers have different corporate governance models?"*, November 2006
Nº 230	Ana Paula Delgado and Isabel Maria Godinho, *"Long term evolution of the size distribution of Portuguese cities"*, September 2006
Nº 229	Sandra Tavares Silva and Aurora A. C. Teixeira, *"On the divergence of evolutionary research paths in the past fifty years: a comprehensive bibliometric account"*, September 2006
Nº 228	Argentino Pessoa, *"Public-Private Sector Partnerships in Developing Countries: Prospects and Drawbacks"*, September 2006
Nº 227	Sandra Tavares Silva and Aurora A. C. Teixeira, *"An evolutionary model of firms' institutional behavior focusing on labor decisions"*, August 2006
Nº 226	Aurora A. C. Teixeira and Natércia Fortuna, *"Human capital, trade and long-run productivity. Testing the technological absorption hypothesis for the Portuguese economy, 1960-2001"*, August 2006
Nº 225	Catarina Monteiro and Aurora A. C. Teixeira, *"Local sustainable mobility management. Are Portuguese municipalities aware?"*, August 2006
Nº 224	Filipe J. Sousa and Luís M. de Castro, *"Of the significance of business relationships"*, July 2006
Nº 223	Pedro Cosme da Costa Vieira, *"Nuclear high-radioactive residues: a new economic solution based on the emergence of a global competitive market"*, July 2006
Nº 222	Paulo Santos, Aurora A. C. Teixeira and Ana Oliveira-Brochado, *"The 'de-territorialisation of closeness' - a typology of international successful R&D projects involving cultural and geographic proximity"*, July 2006
Nº 221	Manuel M. F. Martins, *"Dilemas macroeconómicos e política monetária: o caso da Zona Euro"*, July 2006
Nº 220	Ana Oliveira-Brochado and F. Vitorino Martins, *"Examining the segment retention problem for the "Group Satellite" case"*, July 2006
Nº 219	Óscar Afonso Rui and Henrique Alves, *"To Deficit or Not to Deficit": Should European Fiscal Rules Differ Among Countries?"*, July 2006
Nº 218	Rui Henrique Alves and Óscar Afonso, *"The "New" Stability and Growth Pact: More Flexible, Less Stupid?"*, July 2006
Nº 217	J Maciej Cieślukowski and Rui Henrique Alves, *"Financial Autonomy of the European Union after Enlargement"*, July 2006
Nº 216	Joao Correia-da-Silva and Carlos Hervés-Beloso, *"Prudent Expectations Equilibrium in Economies with Uncertain Delivery"*, June 2006
Nº 215	Maria Rosário Moreira and Rui Alves, *"How far from Just-in-time are Portuguese firms? A survey of its progress and perception"*, June 2006
Nº 214	Maria Fátima Rocha and Aurora A.C. Teixeira, *"A cross-country evaluation of cheating in academia: is it related to 'real world' business ethics?"*, June 2006
Nº 213	Maria Rosário Moreira and Rui Alves, *"Does Order Negotiation Improve The Job-Shop Workload Control?"*, June 2006
Nº 212	Pedro Cosme da Costa Vieira and Aurora A. C. Teixeira, *"Human capital and corruption: a microeconomic model of the bribes market with democratic contestability"*, May 2006
Nº 211	Ana Teresa Tavares and Aurora A. C. Teixeira, *"Is human capital a significant determinant of Portugal's FDI attractiveness?"*, May 2006

Nº 210	Maria Rosário Moreira and Rui Alves, *A new input-output control order release mechanism: how workload control improves manufacturing operations in a job shop*, April 2006
Nº 209	Patrícia Teixeira Lopes and Lúcia Lima Rodrigues, *Accounting for Financial Instruments: An Analysis of the Determinants of Disclosure in the Portuguese Stock Exchange*, April 2006
Nº 208	Pedro Cosme Costa Vieira, *Energia nuclear: Uma solução para Portugal?*, April 2006
Nº 207	Aurora A.C. Teixeira and Joana Costa, *What type of firm forges closer innovation linkages with Portuguese Universities?*, March 2006
Nº 206	Joao Correia-da-Silva and Carlos Hervés-Beloso, *Rational Expectations Equilibrium in Economies with Uncertain Delivery*, March 2006
Nº 205	Luís Delfim Santos and José Varejão, *Employment, Pay and Discrimination in the Tourism Industry*, February 2006
Nº 204	Carlos F. Alves and Victor Mendes, *Mutual fund flows' performance reaction: does convexity apply to small markets?*, February 2006
Nº 203	Carlos F. Alves and Victor Mendes, *Are mutual fund investors in jail?*, February 2006
Nº 202	Óscar Afonso and Paulo B. Vasconcelos, *Numerical computation for initial value problems in economics*, February 2006
Nº 201	Manuel Portugal Ferreira, Ana Teresa Tavares, William Hesterly and Sungu Armagan, *Network and firm antecedents of spin-offs: Motherhooding spin-offs*, February 2006
Nº 200	Aurora A.C. Teixeira, *Vinte anos (1985-2005) de FEP Working Papers: um estudo sobre a respectiva probabilidade de publicação nacional e internacional*, January 2006
Nº 199	Samuel Cruz Alves Pereira, *Aggregation in activity-based costing and the short run activity cost function*, January 2006
Nº 198	Samuel Cruz Alves Pereira and Pedro Cosme Costa Vieira, *How to control market power of activity centres? A theoretical model showing the advantages of implementing competition within organizations*, January 2006
Nº 197	Maria de Fátima Rocha and Aurora A.C. Teixeira, *College cheating in Portugal: results from a large scale survey*, December 2005
Nº 196	Stephen G. Donald, Natércia Fortuna and Vladas Pipiras, *Local and global rank tests for multivariate varying-coefficient models*, December 2005
Nº 195	Pedro Rui Mazeda Gil, *The Firm's Perception of Demand Shocks and the Expected Profitability of Capital under Uncertainty*, December 2005
Nº 194	Ana Oliveira-Brochado and Francisco Vitorino Martins, *Assessing the Number of Components in Mixture Models: a Review.*, November 2005
Nº 193	Lúcia Paiva Martins de Sousa and Pedro Cosme da Costa Vieira, *Um ranking das revistas científicas especializadas em economia regional e urbana*, November 2005

Editor: Sandra Silva (sandras@fep.up.pt)
Download available at:
http://www.fep.up.pt/investigacao/workingpapers/workingpapers.htm
also in http://ideas.repec.org/PaperSeries.html

www.fep.up.pt

Faculdade de Economia da Universidade do Porto
Rua Dr. Roberto Frias, 4200-464 Porto | Tel. 225 571 100
Tel. 225571100 | www.fep.up.pt