FEP WORKING PAPERS
FEP WORKING PAPERS

Research Work in Progress

n. 279, June 2008

Beam search heuristics for quadratic earliness and tardiness scheduling

Jorge M.S. Valente*

* LIAAD, Faculdade de Economia, Universidade do Porto

U.PORTO
FEP FACULDADE DE ECONOMIA UNIVERSIDADE DO PORTO

Beam search heuristics for quadratic earliness and tardiness scheduling

Jorge M. S. Valente*

LIAAD, Faculdade de Economia, Universidade do Porto, Portugal

June 12, 2008

Abstract

In this paper, we present beam search heuristics for the single machine scheduling problem with quadratic earliness and tardiness costs, and no machine idle time. These heuristics include classic beam search procedures, as well as filtered and recovering algorithms. We consider three dispatching heuristics as evaluation functions, in order to analyse the effect of different rules on the performance of the beam search procedures.

The computational results show that using better dispatching heuristics improves the effectiveness of the beam search algorithms. The performance of the several heuristics is similar for instances with low variability. For high variability instances, however, the detailed, filtered and recovering beam search procedures clearly outperform the best existing heuristic. The detailed beam search algorithm performs quite well, and is recommended for small to medium size instances. For larger instances, however, this procedure requires excessive computation times, and the recovering beam search algorithm then becomes the heuristic of choice.

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

Keywords: scheduling, heuristics, beam search, single machine, quadratic earliness, quadratic tardiness.

Introduction

In this paper, we consider a single machine scheduling problem with quadratic earliness and tardiness costs, and no machine idle time. Scheduling models with both earliness and tardiness penalties are compatible with the just-in-time (JIT) production philosophy. The JIT approach focuses on producing goods only when they are needed, and therefore considers that both earliness and tardiness should be discouraged. Also, a recent trend in industry has been the adoption of supply chain management by many organisations. In this approach, customers and suppliers try to integrate the flow of materials, in order to improve the efficiency of the supply chain and provide a better service to the end user. This change to supply chain management has caused organisations to view early deliveries, in addition to tardy deliveries, as undesirable.

We consider quadratic earliness and tardiness penalties, instead of a linear objective function, in order to penalize more heavily deliveries that are quite early or tardy. This is appropriate for practical settings where non-conformance with the due dates is increasingly undesirable. Moreover, the quadratic penalties avoid schedules in which a single or only a few jobs contribute the majority of the cost, without regard to how the overall cost is distributed. The assumption that no machine idle time is allowed is also actually appropriate for many production settings. In fact, idle time should be avoided when the machine has limited capacity or high operating costs, and 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).

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. Also, let h_j and w_j denote the earliness and tardiness penalties of job J_j, respectively. For a given schedule, the earliness and tardiness of J_j are defined as $E_j = \max\{0, d_j - C_j\}$ and $T_j = \max\{0, C_j - d_j\}$, respectively, where C_j is the completion time of J_j. The objective is then to find a schedule that minimizes the sum of the weighted quadratic earliness and tardiness costs $\sum_{j=1}^{n}\left(h_j E_j^2 + w_j T_j^2\right)$, subject to the constraint that no machine idle time is allowed.

This problem has been previously considered by Valente (2007a) and Valente and Alves (2007). Valente (2007a) developed a lower bounding procedure and a branch-and-bound algorithm, while Valente and Alves (2007) presented several dispatching heuristics, as well as simple improvement procedures. The corresponding problem with linear costs $\sum_{j=1}^{n}(h_j E_j + w_j T_j)$ has also been considered by several authors, and both exact and heuristic approaches have been proposed. Among the exact approaches, lower bounds and branch-and-bound algorithms were presented by Abdul-Razaq and Potts (1988), Li (1997), Liaw (1999) and Valente and Alves (2005c). Among the heuristics, several dispatching rules and beam search algorithms were presented by Ow and Morton (1989) and Valente and Alves (2005b,a).

Problems with a related quadratic objective function have also been previously considered. Schaller (2004) analysed the single machine problem with inserted idle time and a linear earliness and quadratic tardiness $\sum_{j=1}^{n}\left(E_j + T_j^2\right)$ objective function. The no idle time version of this problem was considered by Valente (2007b). The minimization of the quadratic lateness, where the lateness of J_j is defined as $L_j = C_j - d_j$, has also been studied by Gupta and Sen (1983), Sen et al. (1995), Su and Chang (1998) and Schaller (2002). Baker and Scudder (1990) and Hoogeveen (2005) provide excellent surveys of scheduling problems with earliness and tardiness penalties, while Kanet and Sridharan (2000) give a review of scheduling models with inserted idle time.

In this paper, we propose several beam search heuristic procedures. Classic beam search procedures are considered, as well as the more recent filtered and recovering beam search approaches. Beam search heuristics require evaluation functions, which are often derived from dispatching rules. Several dispatching rules have been considered, in order to analyse their effect on the effectiveness of the beam search method. The best-performing beam search versions are then compared with the best of the existing heuristics, and with optimal solutions for some instance sizes. In the following, we first describe the beam search approach, and present the proposed heuristics. The computational results are then reported. Finally, we provide some concluding remarks.

The beam search heuristics

Beam search versions and review

Beam search is a heuristic method for solving combinatorial optimization problems. It consists of a truncated branch-and-bound procedure in which only the most promising nodes at each level of the search tree are kept for further branching, while the remaining nodes are pruned off. The classic beam search algorithm was first applied to artificial intelligence problems by Lowerre (1976) and Rubin (1978). Two variations of the traditional beam search algorithm have since been developed. Ow and Morton (1988, 1989) proposed a technique denoted by filtered beam search. Recently, the recovering beam search approach was introduced by Della Croce and T'kindt (2002) and Della Croce et al. (2004).

Beam search heuristics have been applied to several combinatorial optimization problems, with a particular emphasis on the scheduling field. Some recent applications of beam search procedures to scheduling include Della Croce and T'kindt (2002), Della Croce et al. (2004), Valente and Alves (2005a), Ghirardi and Potts (2005) and Esteve et al. (2006). In the following subsections, we present the classic beam search technique, as well as the filtered and recovering variations. We also describe the proposed beam search algorithms, and provide their implementation details.

Classic beam search

The classic beam search procedure consists of a truncated branch-and-bound algorithm in which only the most promising β nodes are kept for further branching at each level of the search tree; β is the so-called *beam width*. The remaining nodes are discarded, and backtracking is not allowed. Therefore, the node evaluation process is crucial for the effectiveness of a beam search algorithm. Two different types of evaluation functions have been used in classic beam search: *priority evaluation functions* and *total cost evaluation functions*.

Priority evaluation functions simply calculate an urgency rating for the last job added to the current partial schedule, typically by using the priority index of a dispatching heuristic. Total cost evaluation functions calculate an estimate of the minimum total cost of the best solution that can be reached from the current node. This is usually done by using a dispatching rule to sequence the unscheduled jobs. Priority evaluation functions have a local view of the problem, because they only consider the next decision to be made, while total cost evaluation functions have a global view, since they project from the current partial solution to a complete schedule.

The priority evaluation functions can pose a slight problem. The priority index that is used to calculate the urgency rating of the last scheduled job usually depends on the current partial schedule (e.g., on the current time). Therefore, the urgency ratings are context-dependent. This means that the priorities calculated for the offspring of one node cannot be legitimately compared with those obtained from the branching of another node. This problem can be solved by initially selecting the best β children of the root node. Then, at lower levels of the search tree, only the best descendant of each beam node is retained for further branching. Total cost evaluation functions are not affected by this problem, since total cost estimates are context-independent and can be compared.

We now present the main steps of priority beam search (PBS) and detailed beam search (DBS) algorithms. The priority (detailed) beam search procedure uses a priority (total cost) evaluation function. In the following,

B is the set of beam nodes , C is a set of offspring nodes and n_0 is the root node.

Priority Beam Search:

Step 1. Initialization:

Set $B = \varnothing$, $C = \varnothing$.

Branch n_0, generating the corresponding children.

Calculate the priority of the last scheduled job for each child node.

Select the best β child nodes and add them to B.

Step 2. Node selection:

For each node in B:

(a) Branch the node, generating the corresponding children.

(b) Calculate the priority of the last scheduled job for each child node.

(c) Select the best child node and add it to C.

Set $B = C$ and $C = \varnothing$.

Step 3. Stopping condition:

If the nodes in B are leaf (i.e., they hold a complete sequence), select the node with the lowest total cost as the best sequence found and stop.

Otherwise, go to step 2.

Detailed Beam Search:

Step 1. Initialization:

Set $B = \{n_0\}$ and $C = \varnothing$.

Step 2. Branching:

For each node in B:

(a) Branch the node, generating the corresponding children.

(b) Calculate an upper bound on the optimal solution value for each child node.

(c) Select the best β child nodes and add them to C.

Set $B = \varnothing$.

Step 3. Node selection:

Select the best β nodes in C and add them to B.

Set $C = \varnothing$.

Step 4. Stopping condition:

If the nodes in B are leaf, select the node with the lowest total cost as the best sequence found and stop.

Otherwise, go to step 2.

Filtered and recovering beam search

The priority evaluation functions are quick, but are rather crude and potentially inaccurate, so they may lead to the elimination of good solutions. Total cost evaluation functions, on the other hand, are more accurate, but much more time consuming. The filtered and recovering beam search algorithms combine crude and accurate evaluations, in order to try to achieve high quality evaluations within reasonable computation times. This is done by means of a two-stage approach. First, a computationally inexpensive *filtering step* is applied. In this step, a crude evaluation is performed, and a reduced number of the offspring of each beam node is selected. These chosen nodes are then accurately evaluated, and the best β are kept for further branching.

Two different types of filtering step have been used. In the approach proposed by Ow and Morton (1988, 1989), a priority evaluation function is used to calculate an urgency rating for each offspring. The best α children of each beam node are then selected for accurate evaluation; α is the so-called *filter width*. The second type of filtering phase was recently introduced by Della Croce and T'kindt (2002) and Della Croce et al. (2004). In this approach, problem-dependent dominance conditions (when available) are applied together with so-called pseudo-dominance conditions (which hold in a heuristic context only). Whenever one of these conditions holds for a given node, that node is eliminated.

The recovering beam search (RBS) approach differs from the filtered beam search (FBS) algorithm in two major ways. First, the accurate evaluation in the filtered beam search procedure relies on an upper bound on the total cost of the best solution that can be reached from the current node. In RBS algorithms, on the other hand, the accurate evaluation uses both lower and upper bounds. More specifically, each node is evaluated by the function $V = (1-\gamma)\,LB + \gamma UB$, where $0 \leq \gamma \leq 1$ is the upper bound weight parameter and LB and UB are the lower and upper bound values, respectively.

Second, the RBS procedure includes a so-called *recovering phase*. In this phase, the nodes that passed the filtering step are considered in non-decreasing order of their evaluation function. For each node, the recovering step then checks if the current partial schedule σ is dominated by another partial schedule $\overline{\sigma}$ with the same level of the search tree. This is typically done by applying neighbourhood operators. If a better partial schedule $\overline{\sigma}$ exists, then σ is replaced by $\overline{\sigma}$. If the possibly modified node is not already in the set of beam nodes, then the node is added to B. This is repeated until either β nodes have been selected, or no additional candidate node remains.

Classic and filtered beam search algorithms cannot recover from wrong decisions: if a node leading to the optimal solution is pruned, there is no way to reach that solution afterwards. The recovering phase seeks to overcome this problem, and often allows the RBS procedure to recover from previous incorrect decisions. We now present the main steps of both filtered and recovering beam search algorithms. In the RBS algorithm, let n_{best} and UB_{best} denote the current best node and the current best upper bound, respectively.

Filtered Beam Search:

Step 1. Initialization:

Set $B = \{n_0\}$ and $C = \varnothing$.

Step 2. Filtering step:

For each node in B:

(a) Branch the node, generating the corresponding children.

(b) Add to C all the child nodes that are not eliminated by the filtering procedure.

Set $B = \varnothing$.

Step 3. Node selection:

Calculate an upper bound on the optimal solution value for all nodes in C.

Select the best β nodes in C and add them to B.

Set $C = \varnothing$.

Step 4. Stopping condition:

If the nodes in B are leaf, select the node with the lowest total cost as the best sequence found and stop.

Otherwise, go to step 2.

Recovering Beam Search:

Step 1. Initialization:

Set $B = \{n_0\}$, $C = \varnothing$, $n_{best} = \varnothing$ and $UB_{best} = \infty$.

Step 2. Filtering step:

For each node in B:

(a) Branch the node, generating the corresponding children.

(b) Add to C all the child nodes that are not eliminated by the filtering procedure.

Set $B = \varnothing$.

Step 3. Accurate evaluation:

For all nodes $n_k, k = 1, \ldots, |C|$ in C:

(a) Calculate a lower bound LB_k and an upper bound UB_k on the optimal solution value of node n_k.

(b) Compute the evaluation function $V_k = (1 - \gamma)\, LB_k + \gamma UB_k$.

(c) If $UB_k < UB_{best}$, set $n_{best} = n_k$ and $UB_{best} = UB_k$.

Step 4. Recovering step:

Sort all nodes in C in non-decreasing order of the evaluation function value V_k.

Set $k = 1$.

While $|B| < \beta$ and $k \leq |C|$:

(a) Let σ represent the partial solution associated with the current node n_k.

(b) Search for a partial solution $\overline{\sigma}$ that dominates σ by means of neighbourhood operators.

(c) If $\overline{\sigma}$ is found, set $\sigma = \overline{\sigma}$.

(d) If $n_k \notin B$

  i. Set $B = B \cup \{n_k\}$.

  ii. If $UB_k < UB_{best}$, set $n_{best} = n_k$ and $UB_{best} = UB_k$.

(e) Set $k = k + 1$.

Step 5. Stopping condition:

If the nodes in B are leaf, stop with n_{best} and UB_{best} as the best node and lowest total cost found, respectively.

Otherwise, go to step 2.

Implementation details

In this paper, we consider both priority and detailed classic beam search algorithms, as well as filtered and recovering beam search procedures. In order to apply these algorithms to the quadratic earliness and tardiness problem, it is necessary to specify their main components, such as branching scheme, evaluation functions, filtering procedure and recovering step. In the following, we provide the implementation details of the beam search heuristics.

Branching scheme

The branching scheme is identical for all algorithms. A forward branching procedure is used, so the sequence is constructed by adding one job at a time, starting from the first position. Therefore, a branch at level l of the search tree indicates the job scheduled in position l.

Dispatching rules

Beam search heuristics require a dispatching rule to calculate upper bounds and/or to provide a priority evaluation function. We considered three alternative dispatching heuristics, in order to analyse their effect of the effectiveness of the beam search procedures. These three heuristics are the EDD, ECTL_AS and ETP_v2 dispatching rules presented in Valente and Alves (2007). The EDD rule is quite well-known and widely used in practice. The ECTL_AS procedure combines the EDD rule with two other simple heuristics, and provides a significant improvement over these simpler rules. The ETP_v2 rule provided the best results of the heuristics analysed in Valente and Alves (2007). Three versions (corresponding to these three rules) were then considered for each type of beam search algorithm. In the following, the ECTL_AS and ETP_v2 rules will be denoted simply as ECTL and ETP, respectively.

Priority beam search

Priority beam search algorithms require a priority evaluation function to calculate the urgency rating of the last scheduled job. This priority function is provided by the priority index of the appropriate dispatching rule (EDD, ECTL or ETP).

Detailed beam search

Detailed beam search algorithms require a total cost evaluation function, i.e., an upper bounding procedure. This procedure is used to sequence the remaining jobs, in order to obtain an upper bound on the total cost of the current partial schedule. The upper bounding procedure is provided by the appropriate dispatching heuristic.

Filtered beam search

Filtered beam search algorithms require a filtering procedure and an upper bounding procedure. Just as previously mentioned for the DBS algorithms, the upper bounding procedure is provided by the relevant dispatching rule. The filtering step uses a priority evaluation function filter, so a priority evaluation function is needed to calculate an urgency rating for the offsprings of a given node. This priority evaluation function is given by the priority index of the appropriate dispatching heuristic, just as previously described for the PBS algorithms.

Recovering beam search

Recovering beam search algorithms require a filtering procedure, upper and lower bounding procedures for the accurate evaluation step, and an improvement procedure for the recovering phase. The filtering and upper bounding procedures are identical to those used in the FBS algorithms. The lower bounding procedure is provided by the method proposed in Valente (2007a). This procedure is used to calculate a lower bound for the unscheduled jobs, and the lower bound of the node is then equal to the sum of the cost of the existing partial schedule and the lower bound calculated for the unscheduled jobs.

Several simple improvement steps for the single machine quadratic earliness and tardiness problem were analysed in Valente and Alves (2007). The adjacent pairwise interchange (API) and 3-swaps (3SW) methods were recommended, since they were both effective and computationally efficient. Therefore, these two improvement procedures were considered for the recovering step in the recovering beam search heuristics.

Improvement step

In the next section, the beam search procedures are compared with the best existing heuristic, as well as with optimum objective function values. In Valente and Alves (2007), the ETP dispatching rule, followed by a 3SW or API improvement step, is recommended as the heuristic procedure of choice. Therefore, we decided to compare the beam search algorithms with the ETP rule with a 3SW improvement step. Consequently, the 3SW method was also applied, as an improvement step, to the beam search procedures (i.e., the 3SW method is used to improve the schedule generated by the beam search heuristics).

Computational results

In this section, we first present the set of test problems used in the computational tests. Then, the preliminary computational experiments are described.

These initial experiments were conducted for two reasons. First, these experiments were performed to determine appropriate values for the parameters required by the several beam search heuristics. Second, these preliminary tests were used to study the performance of the beam search procedures under the EDD, ECTL and ETP rules, in order to select the best-performing. Finally, the computational results are presented. We first compare the beam search heuristics with the best existing procedure, and the heuristic results are then evaluated 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.

Experimental design

The computational tests were performed on a set of problems with 10, 15, 20, 25, 30, 40, 50, 75, 100, 250, 500 and 750 jobs. These problems were randomly generated as follows. For each job J_j, an integer processing time p_j, an integer earliness penalty h_j and an integer tardiness penalty w_j were generated from one of the two uniform distributions $[45, 55]$ and $[1, 100]$, to create low (L) and high (H) variability, respectively. 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 and penalty variability (var), T and R, 50 instances were randomly generated. Therefore, a total of 1200 instances were generated for each combination of problem size and variability. 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 filtered and recovering procedures were not applied to the 750 job instances, and the detailed beam search algorithm was only used on instances with up to 100 jobs.

Preliminary tests

In this section, we describe the preliminary computational experiments. These experiments were conducted, on the one hand, to determine adequate values for the various beam search parameters and, on the other hand, to select the best-performing of the three alternative heuristic rules. A separate problem set was used to conduct these preliminary experiments. This test set included instances with 25, 50, 75 and 100 jobs, and contained 5 instances for each combination of instance size, processing time and penalty variability, T and R. The instances in this smaller test set were generated randomly just as previously described for the full problem set.

We first performed extensive tests to determine appropriate values for the beam width, filter width and upper bound weight parameters. The following values were considered for these parameters: $\alpha = \{1, 2, \ldots, 10\}$, $\beta = \{1, 2, \ldots, 8\}$ and $\gamma = \{0.1, 0.2, \ldots, 0.9\}$. As previously mentioned, the API and 3SW improvement procedures were also considered for the recovering step in the RBS algorithms. The several beam search versions were then applied to the test instances for all combinations of the relevant parameters and improvement procedures. The mean objective function values and run-times were then calculated and plotted. A thorough analysis of these results showed usual behaviour in beam search algorithms: the computation time increases linearly with α and β, while the solution quality improves, but with diminishing returns. The parameter values and improvement procedure that provided the best trade-off between solution quality and computation time were then selected. A value of 3 was chosen for both the beam and filter width parameters, for all beam search versions. In the RBS algorithms, the upper bound weight was set at 0.8, and the API method was selected for the recovering phase.

The performance of the three alternative dispatching heuristics (EDD, ECTL and ETP) was also analysed in these initial experiments, in order to select the best-performing rule. Table 1 presents, for each beam search algorithm, the average of the relative improvements in objective function value over the EDD rule (%imp), as well as the percentage number of times a rule achieves the best objective function value found when compared with the other rules (%best). The relative improvement over the EDD rule is calculated as (edd_ofv - rule_ofv) / edd_ofv $\times$ 100, where edd_ofv and rule_ofv are the objective function values obtained by the EDD rule and the appropriate rule (ECTL or ETP), respectively. These values are omitted for the EDD rule, since they would all be necessarily equal to 0.

The objective function values provided by the EDD, ECTL and ETP rules are close for the instances with low variability. Indeed, the relative improvement given by the ECTL and ETP heuristics is less than 1% for the DBS and FBS procedures, and negligible for the RBS algorithm. For the

PBS procedure, the relative improvement is a little higher (around 1.5%). Nevertheless, the ECTL and ETP rules provide the best results for a much larger number of instances. The ETP rule, in particular, provides the best results for over 90%, and in some cases actually all, of the test instances. For the high variability instances, the ECTL and (especially) the ETP rules are greatly superior to the EDD heuristic. In fact, the ECTL and ETP rules provide a quite large relative improvement, and also give the best results for a much higher percentage of the test instances.

The relative improvement provided by the ECTL and ETP rules is higher for the PBS algorithm, which only uses priority evaluation. The improvement is smaller for the FBS (both priority and detailed evaluations) and DBS (detailed evaluation only) procedures, but the more advanced ECTL and ETP rules still provide a quite large improvement for the instances with high variability. Therefore, it certainly seems that high quality rules should be used to provide both priority evaluation functions and upper bounding procedures in beam search heuristics for the considered scheduling problem. The objective function values given by the three rules are closer for the RBS algorithm, which is most likely due to the recovering phase. Indeed, incorrect choices made by an inferior rule can later be corrected by the recovering step, and so the results provided by the alternative rules are closer.

The ETP rule was then selected, since it proved superior to its alternatives, particularly for the instances with a high variability. Therefore, in the following sections we will present results only for the ETP versions of the beam search heuristics.

Heuristic results

In this section, the beam search algorithms are compared with the best of the existing procedures, namely the ETP dispatching rule. As previously mentioned, the 3SW method is used as an improvement step, in order to improve the schedules generated by the several heuristics. In table 2, we provide the average of the relative improvements in objective function value over the ETP procedure (%imp), as well as the percentage number of times a heuristic achieves the best result when compared with the other heuristics (%best). The relative improvement over the ETP heuristic is calculated as (etp_ofv - heur_ofv) / etp_ofv $\times$ 100, where heur_ofv and etp_ofv are the objective function values of the appropriate heuristic and the ETP dispatching rule, respectively. The relative improvement values are omitted for the ETP heuristic, since they are necessarily equal to 0.

The performance of the several beam algorithms and the ETP dispatching rule is virtually identical for the instances with low variability. Indeed, the objective function values are quite close, and all the heuristics provide the best results for over 90% of the test instances. For instances with high variability, however, the DBS, FBS and RBS procedures are clearly superior to the dispatching heuristic. In fact, these procedures give a relative improvement that ranges from 1% to 3%, and provide the best results for a larger number of instances.

The best results are given by the DBS procedure, closely followed by the RBS algorithm. The FBS algorithm, though clearly superior to the ETP heuristic, is outperformed by the DBS and RBS procedures. On the one hand, the DBS algorithm applies a detailed evaluation to all nodes, which can account for its superior performance. On the other hand, the RBS heuristic not only uses a weighted average of lower and upper bounds in its detailed evaluation, but also benefits from the local search that is performed in the recovering phase. The PBS procedure only provides a minor relative improvement over the ETP dispatching rule, and the percentage of best results is also quite close for these two heuristics.

Table 3 presents the effect of the T and R parameters on the relative improvement over the ETP dispatching rule, for instances with 50 jobs. The relative improvement is quite minor when $T = 0.0$ or $T = 1.0$. However, the improvement is quite significant for the intermediate values of the tardiness factor (and also for instances with $T = 0.8$ and a small due date range). This result is to be expected, since the heuristics are more likely to be closer to the optimum for extreme values of the tardiness factor T. Indeed, when $T = 0.0$ ($T = 1.0$), most jobs will be early (late), and the early/tardy scheduling problem is easier. For the intermediate values of the tardiness factor, there is a greater balance between the number of early and tardy jobs, and the problem then becomes harder.

The heuristic runtimes (in seconds) are presented in table 4. The DBS procedure is computationally quite demanding, and can only be used for small or medium size instances. The FBS and RBS algorithms are faster, and can be applied to somewhat larger instances. The PBS procedure is much faster than the other beam search algorithms. However, the ETP dispatching rule is even more computationally efficient, and provides results of similar quality. The DBS procedure is then recommended for small to medium instance sizes. For medium to large instances, the RBS heuristic is the procedure of choice. The ETP dispatching rule is quite computationally efficient, and is the only procedure that can provide results in reasonable times for very large instances.

Comparison with optimum results

In this section, we compare the heuristic results with optimum objective function values, for instances with up to 20 jobs. Table 5 presents the average of the relative deviations from the optimum (%dev), calculated as $(H - O)/O \times 100$, 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.

The heuristic procedures perform extremely well for the instances with low variability. Indeed, all the heuristics provide the optimal solution value for over 96% of these instances. The differences in performance are much clearer for the high variability instances. The DBS and RBS algorithms still perform quite well, since they give results that are about 1% above the optimum, and provide an optimum solution for over 80% of the instances. The performance of the FBS algorithm is also quite good, as its average deviation from the optimum is around 1-2%. The PBS and ETP heuristics perform adequately, but are clearly outperformed by the DBS, RBS and FBS procedures. In fact, the PBS (ETP) heuristic provides results that are about 3-4% (5-6%) above the optimum.

These results are in accordance with those presented in the previous section. In fact, as previously mentioned, the performance of the heuristic procedures was virtually identical for the low variability instances. For instances with high variability, however, the DBS, FBS and RBS heuristics were clearly superior to the ETP dispatching heuristic. We can now see that the ETP heuristic is nearly always optimal for the low variability instances, so there was nearly no room for improvement. For the instances with high variability, however, the performance of the ETP heuristic deteriorates, and the beam search algorithms can therefore achieve a larger improvement.

The effect of the T and R parameters on the relative deviation from the optimum is presented in table 6, for instances with 20 jobs. The heuristic procedures are quite close to the optimum for the extreme values of T, but their performance deteriorates for the intermediate values of the tardiness factor. Therefore, the heuristics are nearly optimal when most jobs are early or tardy, and their performance worsens as the number of early and tardy jobs becomes more balanced. Again, these results are in line with those reported in the previous section.

Conclusion

In this paper, we proposed several beam search heuristics for the single machine scheduling problem with quadratic earliness and tardiness costs, and no machine idle time. These heuristics included classic procedures, and also filtered and recovering algorithms. Beam search procedures require evaluation functions, and these are usually derived from dispatching heuristics. We considered three alternative dispatching rules, in order to analyse their effect on the performance of the beam search algorithms.

The preliminary computational experiments show that using better dispatching rules indeed improves the performance of the beam search algorithms, especially for the instances with high processing time and penalty variability. The best-performing beam search versions were then compared with the ETP dispatching rule (the best existing heuristic) and with optimal solutions. The computational results show that all the heuristic procedures perform extremely well when the variability is low, generating an optimal solution for over 96% of these instances. The difference in performance is much clearer for the harder high variability instances, where the DBS, RBS and FBS algorithms are clearly superior to the best existing procedure. The DBS heuristic performs quite well, and is recommended for small to medium size instances. For larger instances, however, this procedure requires excessive computation times, and the RBS algorithm is then the heuristic of choice.

References

Abdul-Razaq T and Potts C N (1988). Dynamic programming state-space relaxation for single machine scheduling. Journal of the Operational Research Society 39: 141–152.

Baker K R and Scudder G D (1990). Sequencing with earliness and tardiness penalties: A review. Operations Research 38: 22–36.

Della Croce F, Ghirardi M and Tadei R (2004). Recovering beam search:

Enhancing the beam search approach for combinatorial problems. Journal of Heuristics 10: 89–104.

Della Croce F and T'kindt V (2002). A recovering beam search algorithm for the one-machine dynamic total completion time scheduling problem. Journal of the Operational Research Society 53: 1275–1280.

Esteve B, Aubijoux C, Chartier A and T'kindt V (2006). A recovering beam search algorithm for the single machine just-in-time scheduling problem. European Journal of Operational Research 172: 798–813.

Ghirardi M and Potts C N (2005). Makespan minimization for scheduling unrelated parallel machines: A recovering beam search approach. European Journal of Operational Research 165: 457–467.

Gupta S K and Sen T (1983). Minimizing a quadratic function of job lateness on a single machine. Engineering Costs and Production Economics 7: 187–194.

Hoogeveen H (2005). Multicriteria scheduling. European Journal of Operational Research 167: 592–623.

Kanet J J and Sridharan V (2000). Scheduling with inserted idle time: Problem taxonomy and literature review. Operations Research 48: 99–110.

Korman K (1994). A pressing matter. Video : 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.

Li G (1997). Single machine earliness and tardiness scheduling. European Journal of Operational Research 96: 546–558.

Liaw C F (1999). A branch-and-bound algorithm for the single machine earliness and tardiness scheduling problem. Computers & Operations Research 26: 679–693.

Lowerre B T (1976). The HARPY Speech Recognition System. Ph.d. thesis, Carnegie-Mellon University, USA.

Ow P S and Morton T E (1988). Filtered beam search in scheduling. International Journal of Production Research 26: 35–62.

Ow P S and Morton T E (1989). The single machine early/tardy problem. Management Science 35: 177–191.

Rubin S (1978). The ARGOS Image Understanding System. Ph.d. thesis, Carnegie-Mellon University, USA.

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, Dileepan P and Lind M R (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 Chang P C (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 (2007a). An exact approach for single machine scheduling with quadratic earliness and tardiness penalties. Working Paper 238, Faculdade de Economia, Universidade do Porto, Portugal.

Valente J M S (2007b). Heuristics for the single machine scheduling problem with early and quadratic tardy penalties. European Journal of Industrial Engineering 1: 431–448.

Valente J M S and Alves R A F S (2005a). Filtered and recovering beam search algorithms for the early/tardy scheduling problem with no idle time. Computers & Industrial Engineering 48: 363–375.

Valente J M S and Alves R A F S (2005b). Improved heuristics for the early/tardy scheduling problem with no idle time. Computers & Operations Research 32: 557–569.

Valente J M S and Alves R A F S (2005c). Improved lower bounds for the early/tardy scheduling problem with no idle time. Journal of the Operational Research Society 56: 604–612.

Valente J M S and Alves R A F S (2007). Heuristics for the single machine scheduling problem with quadratic earliness and tardiness penalties. Working Paper 236, Faculdade de Economia do Porto, Portugal (to appear in Computers & Operations Research).

			$n = 25$		$n = 50$		$n = 75$		$n = 100$	
var	heur	rule	%imp	%best	%imp	%best	%imp	%best	%imp	%best
L	PBS	EDD	—	0.83	—	0.00	—	0.00	—	0.00
		ECTL	1.37	71.67	1.45	54.17	1.58	50.83	1.26	47.50
		ETP	1.41	96.67	1.65	99.17	1.69	100.00	1.65	99.17
	DBS	EDD	—	14.17	—	0.83	—	0.00	—	0.00
		ECTL	0.44	92.50	0.67	71.67	0.81	56.67	0.81	58.33
		ETP	0.44	98.33	0.70	99.17	0.84	100.00	0.91	100.00
	FBS	EDD	—	18.33	—	0.00	—	0.00	—	0.00
		ECTL	0.32	95.00	0.60	73.33	0.79	57.50	0.76	56.67
		ETP	0.32	99.17	0.63	100.00	0.81	100.00	0.90	100.00
	RBS	EDD	—	95.00	—	89.17	—	77.50	—	79.17
		ECTL	0.00	99.17	0.00	92.50	0.00	86.67	0.00	92.50
		ETP	0.00	100.00	0.00	95.00	0.00	92.50	0.00	95.83
H	PBS	EDD	—	0.00	—	0.00	—	0.00	—	0.00
		ECTL	50.42	43.33	51.71	33.33	53.41	33.33	53.53	40.00
		ETP	54.65	90.00	56.68	98.33	58.55	97.50	58.78	96.67
	DBS	EDD	—	2.50	—	0.00	—	0.00	—	0.83
		ECTL	17.59	54.17	22.18	44.17	24.43	44.17	25.70	44.17
		ETP	19.07	88.33	24.96	93.33	26.77	92.50	28.01	93.33
	FBS	EDD	—	0.00	—	0.00	—	0.00	—	0.00
		ECTL	35.68	52.50	43.05	45.00	47.73	40.83	49.49	42.50
		ETP	37.40	90.83	45.56	92.50	50.67	95.00	52.83	97.50
	RBS	EDD	—	48.33	—	31.67	—	30.83	—	29.17
		ECTL	3.64	70.83	6.19	50.83	5.69	39.17	6.17	44.17
		ETP	5.50	85.83	7.93	75.83	7.09	64.17	7.71	67.50

Table 1: Preliminary results

		$n = 25$		$n = 50$		$n = 100$		$n = 500$	
var	heur	%imp	%best	%imp	%best	%imp	%best	%imp	%best
L	ETP	—	96.25	—	92.33	—	91.50	—	95.42
	PBS	0.000	96.33	0.000	92.33	0.000	91.50	0.000	95.42
	DBS	0.002	98.92	0.001	97.75	0.000	96.67	—	—
	FBS	0.002	98.92	0.001	97.08	0.000	95.92	0.000	98.17
	RBS	0.002	99.67	0.001	98.42	0.000	96.67	0.000	97.67
H	ETP	—	61.75	—	48.17	—	38.00	—	37.33
	PBS	0.423	63.00	0.109	48.08	0.010	37.92	0.004	37.33
	DBS	3.092	86.17	2.311	79.17	1.626	72.25	—	—
	FBS	2.233	76.50	1.350	63.08	0.795	52.50	0.160	67.67
	RBS	2.973	88.25	2.089	74.58	1.389	62.17	0.396	66.42

Table 2: Heuristic results

heur	T	low var $R = 0.2$	$R = 0.4$	$R = 0.6$	$R = 0.8$	high var $R = 0.2$	$R = 0.4$	$R = 0.6$	$R = 0.8$
PBS	0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.6	0.000	0.000	0.000	0.000	0.000	0.000	0.247	0.475
	0.8	0.000	0.000	0.000	0.000	1.279	0.609	0.000	0.000
	1.0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
DBS	0.0	0.001	0.000	0.000	0.000	0.000	0.001	0.001	0.003
	0.2	0.001	0.000	0.000	0.000	0.137	0.068	0.082	0.032
	0.4	0.006	0.001	0.000	0.000	4.083	1.906	2.685	1.263
	0.6	0.001	0.002	0.001	0.000	8.095	9.205	9.919	6.654
	0.8	0.002	0.000	0.000	0.000	10.000	1.243	0.048	0.038
	1.0	0.000	0.000	0.000	0.000	0.006	0.002	0.003	0.002
FBS	0.0	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.003
	0.2	0.001	0.000	0.000	0.000	0.078	0.056	0.061	0.003
	0.4	0.005	0.001	0.000	0.000	2.169	1.383	1.616	0.831
	0.6	0.001	0.002	0.001	0.000	4.094	3.516	6.383	4.770
	0.8	0.001	0.000	0.000	0.000	6.530	0.832	0.033	0.022
	1.0	0.000	0.000	0.000	0.000	0.005	0.002	0.003	0.001
RBS	0.0	0.000	0.000	0.000	0.000	0.000	0.003	0.001	0.003
	0.2	0.001	0.000	0.000	0.000	0.147	0.127	0.104	0.022
	0.4	0.006	0.001	0.000	0.000	4.196	2.150	3.217	2.277
	0.6	0.006	0.002	0.001	0.000	7.632	7.906	8.105	6.102
	0.8	0.003	0.000	0.000	0.000	7.194	0.838	0.046	0.050
	1.0	0.001	0.000	0.000	0.000	0.004	0.001	0.000	0.000

Table 3: Relative improvement over the ETP heuristic, for instances with 50 jobs

var	heur	$n = 25$	$n = 50$	$n = 75$	$n = 100$	$n = 250$	$n = 500$
L	ETP	0.000	0.000	0.001	0.001	0.004	0.013
	PBS	0.002	0.006	0.014	0.026	0.209	2.919
	DBS	0.015	0.206	1.022	3.197	—	—
	FBS	0.004	0.023	0.068	0.154	2.472	20.803
	RBS	0.007	0.037	0.104	0.225	3.240	25.866
H	ETP	0.000	0.000	0.001	0.001	0.004	0.014
	PBS	0.002	0.007	0.015	0.027	0.208	2.820
	DBS	0.016	0.214	1.041	3.302	—	—
	FBS	0.004	0.024	0.072	0.166	2.545	21.678
	RBS	0.007	0.038	0.109	0.240	3.381	27.547

Table 4: Heuristic runtimes (in seconds)

		$n = 10$		$n = 15$		$n = 20$	
var	heur	%dev	%opt	%dev	%opt	%dev	%opt
L	ETP	0.007	98.50	0.002	97.92	0.002	96.58
	PBS	0.005	98.58	0.002	98.00	0.002	96.67
	DBS	0.001	99.42	0.000	99.50	0.000	99.17
	FBS	0.001	99.33	0.000	99.58	0.001	98.83
	RBS	0.000	99.92	0.000	100.00	0.000	99.67
H	ETP	4.690	80.75	5.168	70.67	5.892	64.83
	PBS	2.862	83.92	3.878	72.17	4.832	65.75
	DBS	0.366	95.33	0.737	86.50	1.103	80.67
	FBS	0.378	93.42	1.380	83.08	2.309	76.00
	RBS	0.221	95.67	0.907	88.00	1.397	82.50

Table 5: Comparison with optimum objective function values

		low var				high var			
heur	T	$R = 0.2$	$R = 0.4$	$R = 0.6$	$R = 0.8$	$R = 0.2$	$R = 0.4$	$R = 0.6$	$R = 0.8$
ETP	0.0	0.000	0.000	0.000	0.000	0.005	0.054	0.014	0.000
	0.2	0.032	0.000	0.000	0.000	0.538	0.251	0.277	0.091
	0.4	0.003	0.000	0.000	0.000	13.353	14.822	9.408	10.565
	0.6	0.011	0.000	0.003	0.000	33.873	20.353	12.395	10.053
	0.8	0.003	0.000	0.000	0.001	10.274	3.765	0.852	0.356
	1.0	0.001	0.000	0.000	0.000	0.040	0.030	0.030	0.011
PBS	0.0	0.000	0.000	0.000	0.000	0.005	0.054	0.014	0.000
	0.2	0.032	0.000	0.000	0.000	0.538	0.251	0.277	0.091
	0.4	0.003	0.000	0.000	0.000	13.353	14.822	8.397	6.341
	0.6	0.011	0.000	0.000	0.000	30.740	14.926	9.637	9.941
	0.8	0.003	0.000	0.000	0.001	3.394	1.894	0.824	0.356
	1.0	0.001	0.000	0.000	0.000	0.040	0.030	0.030	0.011
DBS	0.0	0.000	0.000	0.000	0.000	0.000	0.034	0.000	0.000
	0.2	0.004	0.000	0.000	0.000	0.289	0.085	0.148	0.033
	0.4	0.000	0.000	0.000	0.000	3.969	6.137	1.835	1.097
	0.6	0.002	0.000	0.000	0.000	7.645	3.212	1.235	0.349
	0.8	0.001	0.000	0.000	0.000	0.222	0.152	0.018	0.005
	1.0	0.000	0.000	0.000	0.000	0.001	0.001	0.011	0.000
FBS	0.0	0.000	0.000	0.000	0.000	0.000	0.034	0.000	0.009
	0.2	0.005	0.000	0.000	0.000	0.336	0.085	0.153	0.038
	0.4	0.000	0.000	0.000	0.000	8.706	7.928	2.762	2.155
	0.6	0.006	0.000	0.000	0.000	16.040	6.592	5.176	1.135
	0.8	0.001	0.000	0.000	0.000	2.760	1.337	0.026	0.129
	1.0	0.000	0.000	0.000	0.000	0.001	0.002	0.008	0.000
RBS	0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	0.2	0.000	0.000	0.000	0.000	0.105	0.004	0.000	0.005
	0.4	0.000	0.000	0.000	0.000	3.880	6.219	0.681	0.804
	0.6	0.000	0.000	0.000	0.000	10.943	3.634	2.638	0.735
	0.8	0.000	0.000	0.000	0.000	2.614	1.053	0.031	0.171
	1.0	0.001	0.000	0.000	0.000	0.005	0.002	0.006	0.000

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

Recent FEP Working Papers

Nº 278	Nuno Torres and Óscar Afonso, "*Re-evaluating the impact of natural resources on economic growth*", June 2008
Nº 277	Inês Drumond, "*Bank Capital Requirements, Business Cycle Fluctuations and the Basel Accords: A Synthesis*", June 2008
Nº 276	Pedro Rui Mazeda Gil, "*Stylized Facts and Other Empirical Evidence on Firm Dynamics, Business Cycle and Growth*", May 2008
Nº 275	Teresa Dieguez and Aurora A.C. Teixeira, "*ICTs and Family Physicians Human Capital Upgrading. Delightful Chimera or Harsh Reality?*", May 2008
Nº 274	Teresa M. Fernandes, João F. Proença and P.K. Kannan, "*The Relationships in Marketing: Contribution of a Historical Perspective*", May 2008
Nº 273	Paulo Guimarães, Octávio Figueiredo and Douglas Woodward, "*Dartboard Tests for the Location Quotient*", April 2008
Nº 272	Rui Leite and Óscar Afonso, "*Effects of learning-by-doing, technology-adoption costs and wage inequality*", April 2008
Nº 271	Aurora A.C. Teixeira, "*National Systems of Innovation: a bibliometric appraisal*", April 2008
Nº 270	Tiago Mata, "*An uncertain dollar: The Wall Street Journal, the New York Times and the monetary crisis of 1971 to 1973*", April 2008
Nº 269	João Correia-da-Silva and Carlos Hervés-Beloso, "*General equilibrium with private state verification*", March 2008
Nº 268	Carlos Brito, "*Relationship Marketing: From Its Origins to the Current Streams of Research*", March 2008
Nº 267	Argentino Pessoa, "*Kuznets's Hypothesis And The Data Constraint*", February 2008
Nº 266	Argentino Pessoa, "*Public-Private Sector Partnerships In Developing Countries: Are Infrastructures Responding To The New Oda Strategy*", February 2008
Nº 265	Álvaro Aguiar and Ana Paula Ribeiro, "*Why Do Central Banks Push for Structural Reforms? The Case of a Reform in the Labor Market*", February 2008
Nº 264	Jorge M. S. Valente and José Fernando Gonçalves, "*A genetic algorithm approach for the single machine scheduling problem with linear earliness and quadratic tardiness penalties*", January 2008
Nº 263	Ana Oliveira-Brochado and Francisco Vitorino Martins, *"Determining the Number of Market Segments Using an Experimental Design"*, January 2008
Nº 262	Ana Oliveira-Brochado and Francisco Vitorino Martins, *"Segmentação de mercado e modelos mistura de regressão para variáveis normais"*, January 2008
Nº 261	Ana Oliveira-Brochado and Francisco Vitorino Martins, *"Aspectos Metodológicos da Segmentação de Mercado: Base de Segmentação e Métodos de Classificação"*, January 2008
Nº 260	João Correia-da-Silva, *"Agreeing to disagree in a countable space of equiprobable states"*, January 2008
Nº 259	Rui Cunha Marques and Ana Oliveira-Brochado, *"Comparing Airport regulation in Europe: Is there need for a European Regulator?"*, December 2007
Nº 258	Ana Oliveira-Brochado and Rui Cunha Marques, *"Comparing alternative instruments to measure service quality in higher education"*, December 2007
Nº 257	Sara C. Santos Cruz and Aurora A.C. Teixeira, *"A new look into the evolution of clusters literature. A bibliometric exercise"*, December 2007
Nº 256	Aurora A.C. Teixeira, *"Entrepreneurial potential in Business and Engineering courses … why worry now?"*, December 2007
Nº 255	Alexandre Almeida and Aurora A.C. Teixeira, *"Does Patenting negatively impact on R&D investment?An international panel data assessment"*, December 2007
Nº 254	Argentino Pessoa, *"Innovation and Economic Growth: What is the actual importance of R&D?"*, November 2007
Nº 253	Gabriel Leite Mota, *"Why Should Happiness Have a Role in Welfare Economics? Happiness versus Orthodoxy and Capabilities"*, November 2007

Nº 252	Manuel Mota Freitas Martins, *"Terá a política monetária do Banco Central Europeu sido adequada para Portugal (1999-2007)?"*, November 2007
Nº 251	Argentino Pessoa, *"FDI and Host Country Productivity: A Review"*, October 2007
Nº 250	Jorge M. S. Valente, *"Beam search heuristics for the single machine scheduling problem with linear earliness and quadratic tardiness costs"*, October 2007
Nº 249	T. Andrade, G. Faria, V. Leite, F. Verona, M. Viegas, O.Afonso and P.B. Vasconcelos, *"Numerical solution of linear models in economics: The SP-DG model revisited"*, October 2007
Nº 248	Mário Alexandre P. M. Silva, *"Aghion And Howitt's Basic Schumpeterian Model Of Growth Through Creative Destruction: A Geometric Interpretation"*, October 2007
Nº 247	Octávio Figueiredo, Paulo Guimarães and Douglas Woodward, *"Localization Economies and Establishment Scale: A Dartboard Approach"*, September 2007
Nº 246	Dalila B. M. M. Fontes, Luís Camões and Fernando A. C. C. Fontes, *"Real Options using Markov Chains: an application to Production Capacity Decisions"*, July 2007
Nº 245	Fernando A. C. C. Fontes and Dalila B. M. M. Fontes, *"Optimal investment timing using Markov jump price processes"*, July 2007
Nº 244	Rui Henrique Alves and Óscar Afonso, *"Fiscal Federalism in the European Union: How Far Are We?"*, July 2007
Nº 243	Dalila B. M. M. Fontes, *"Computational results for Constrained Minimum Spanning Trees in Flow Networks"*, June 2007
Nº 242	Álvaro Aguiar and Inês Drumond, *"Business Cycle and Bank Capital: Monetary Policy Transmission under the Basel Accords"*, June 2007
Nº 241	Sandra T. Silva, Jorge M. S. Valente and Aurora A. C. Teixeira, *"An evolutionary model of industry dynamics and firms' institutional behavior with job search, bargaining and matching"*, April 2007
Nº 240	António Miguel Martins and Ana Paula Serra, *"Market Impact of International Sporting and Cultural Events"*, April 2007
Nº 239	Patrícia Teixeira Lopes and Lúcia Lima Rodrigues, *"Accounting for financial instruments: A comparison of European companies' practices with IAS 32 and IAS 39"*, March 2007
Nº 238	Jorge M. S. Valente, *"An exact approach for single machine scheduling with quadratic earliness and tardiness penalties"*, February 2007
Nº 237	Álvaro Aguiar and Ana Paula Ribeiro, *"Monetary Policy and the Political Support for a Labor Market Reform"*, February 2007
Nº 236	Jorge M. S. Valente and Rui A. F. S. Alves, *"Heuristics for the single machine scheduling problem with quadratic earliness and tardiness penalties"*, February 2007
Nº 235	Manuela Magalhães and Ana Paula Africano, *"A Panel Analysis of the FDI Impact on International Trade"*, January 2007
Nº 234	Jorge M. S. Valente, *"Heuristics for the single machine scheduling problem with early and quadratic tardy penalties"*, December 2006

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