FEP WORKING PAPERS
FEP WORKING PAPERS

Research Work in Progress

n. 286, August 2008

Greedy randomized dispatching heuristics for the single machine scheduling problem with quadratic earliness and tardiness penalties

Jorge M. S. Valente
Maria R. A. Moreira

U.PORTO
FEP FACULDADE DE ECONOMIA UNIVERSIDADE DO PORTO

Greedy randomized dispatching heuristics for the single machine scheduling problem with quadratic earliness and tardiness penalties

Jorge M. S. Valente

LIAAD, Faculdade de Economia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-464 Porto, Portugal

Phone: + 351 225 571 100

Fax: + 351 225 505 050

E-mail: jvalente@fep.up.pt

Maria R. A. Moreira

EDGE, Faculdade de Economia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-464 Porto, Portugal

Abstract In this paper, we present greedy randomized dispatching heuristics for the single machine scheduling problem with quadratic earliness and tardiness costs, and no machine idle time. The several heuristic versions differ, on the one hand, on the strategies involved in the construction of the greedy randomized schedules. On the other hand, these versions also differ on whether they employ only a final improvement step, or perform a local search after each greedy randomized construction.

The proposed heuristics were compared with existing procedures, as well as with optimum solutions for some instance sizes. The computational results show that the proposed procedures clearly outperform their underlying dispatching heuristic, and the best of these procedures provide results that are quite close to the optimum. The best of the proposed algorithms is the new recommended heuristic for large instances, as well as a suitable alternative to the best existing procedure for the larger of the middle size instances.

Keywords *scheduling, single machine, early/tardy, quadratic penalties, greedy randomized dispatching rules*

Introduction

Scheduling is concerned with the allocation of scarce resources to activities over time, in order to optimize one or more objectives. In this paper, we consider a single machine production scheduling problem. Therefore, in this context the scarce resource is the machine, while the activities are the several jobs that have to be processed on that machine. More specifically, we consider the 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 production philosophy, which emphasizes 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. This change to supply chain management has also 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. This penalizes more heavily deliveries that are quite early or tardy, which is appropriate for practical settings where non-conformance with the due dates is highly undesirable. 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 [8] and Landis [9].

Formally, the problem can be stated as follows. A set of n independent jobs $\{1, 2, \ldots, n\}$ has to be scheduled on 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, $j = 1, 2, \ldots, n$, 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, respectively. For a given schedule, the earliness and tardiness of job 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 job 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 in [18, 25, 21]. Valente [18] developed a lower bounding procedure and a branch-and-bound algorithm. In [25], Valente and Alves presented several dispatching rules, as well as simple improvement procedures. Valente [21] considered classic, filtered and recovering beam search heuristics. The corresponding problem with linear costs

$\sum_{j=1}^{n}\left(h_j E_j + w_j T_j\right)$ has also been considered by several authors, and both exact [1, 10, 11, 24] and heuristic [12, 23, 22] approaches have been proposed. Problems with a related quadratic objective function have also been previously considered. Schaller [15] 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, while the no idle time version of this problem was considered by Valente [20, 19]. The minimization of the quadratic lateness, where the lateness of job j is defined as $L_j = C_j - d_j$, has also been studied in [5, 16, 17, 14]. Baker and Scudder [2] and Hoogeveen [6] provide excellent surveys of scheduling problems with earliness and tardiness penalties, while a review of scheduling models with inserted idle time is given in [7].

In this paper, we present greedy randomized dispatching heuristics, and analyse their performance on a wide range of instances. Several different heuristic versions are considered. These versions, on the one hand, differ on the strategies involved in the construction of the greedy randomized schedules. On the other hand, the versions also differ on whether they employ only a final improvement step, or perform a local search after each greedy randomized construction. The proposed heuristics are compared with existing procedures, as well as with optimum solutions for some instance sizes.

The remainder of this paper is organized as follows. In the next section, we describe the basic elements of greedy randomized dispatching rules. The proposed heuristics are presented in the following section. Then, the computational results are reported. Finally, we provide some concluding remarks.

Greedy randomized dispatching rules

Dispatching rules are quite popular in scheduling, and a large number of papers in scheduling literature present or analyse dispatching heuristics. Also, dispatching rules are widely used in practice. In fact, most real scheduling systems are either based on dispatching heuristics, or at least use them to some degree. Moreover, dispatching rules are sometimes the only heuristic approach capable of generating a solution within reasonable computation times for large instances. Furthermore, dispatching rules are also frequently used by other heuristic procedures, e.g. they

are often used to generate the initial sequence required by local search or metaheuristic algorithms.

Dispatching rules are constructive heuristics, i.e. they construct a solution one element or component (e.g. one job) at a time. Therefore, at each iteration a feasible element is added to the current partial solution. More specifically, dispatching heuristics typically calculate a priority or urgency rating for each unscheduled job each time the machine becomes available, and the job with the largest priority value is then selected for processing.

These heuristics, as well as similar constructive heuristics, are deterministic, and therefore they are only capable of producing a single schedule. One way to enhance the performance of dispatching heuristics is by introducing randomization. In this context, each time a job is to be chosen, that selection is randomized, so a job different from the one with the largest priority value can be selected for processing. This randomized selection is usually performed in a greedy fashion, so that jobs with larger priorities have a higher probability of being chosen.

Each time such a greedy randomized dispatching rule is used, a different schedule may then be obtained. Therefore, iteratively applying this greedy randomized rule leads to a more extensive search of the solution space. As such, the iterative greedy randomization of a dispatching rule can be an effective way of boosting the performance of dispatching heuristics or similar constructive procedures.

Greedy randomized constructive heuristics are also used in the construction phase of the GRASP metaheuristic [13]. In this metaheuristic, greedy randomized solutions are first constructed, and then improved by a local search procedure, until a stopping criterion is met.

Procedure 1: Greedy Randomized Construction

1. Set $S = \emptyset$ and $U = \{1, 2, \ldots, n\}$.
2. While $U \neq \emptyset$
   - 2.1. Calculate the priority value I_j for all jobs $j \in U$.
   - 2.2. Create a candidate list CL of the unscheduled jobs that will be considered to be scheduled in the next position.
   - 2.3. Calculate the score sc_j for all jobs $j \in CL$.
   - 2.4. Calculate the biased score bsc_j for all jobs $j \in CL$.

2.5. Calculate the probability $prob_j$ of selecting each job $j \in CL$: $prob_j = bsc_j / \Sigma_{j \in CL} bsc_j$.

2.6. Randomly select the next job to be scheduled from the jobs in *CL* according to the probabilities $prob_j$.

2.7. Add the selected job to set *S* and remove it from set *U*.

3. Return the schedule in set *S*.

The pseudo-code for the greedy randomized construction of a schedule for the considered single machine scheduling problem is given in Procedure 1. Several different strategies may be used to perform steps 2.2, 2.3 and 2.4, thereby leading to different greedy randomized heuristic versions.

In step 2.2, the candidate list can simply be set to all the unscheduled jobs in set *U*; this strategy will be denoted by *All*. In the GRASP metaheuristic, on the other hand, the candidate list is typically a subset of the unscheduled jobs, and is denoted by restricted candidate list (*RCL*). In this case, the *RCL* is usually constructed using a user-defined threshold parameter $\alpha \in [0, 1]$. The *RCL* then contains the jobs with priority values $I_j \in [I_{max} - \alpha(I_{max} - I_{min}), I_{max}]$, where I_{max} and I_{min} are the maximum and minimum values of the priorities of all the unscheduled jobs, respectively. When $\alpha = 0$, we obtain the deterministic heuristic, given that only the job with the largest priority value would be included in the *RCL*. On the other hand, the value $\alpha = 1$ corresponds to the strategy *All*, since we would then have $CL = U$.

In step 2.3, a score is calculated for each job in the candidate list. In the heuristic-biased stochastic sampling approach proposed by Bresina [3], and denoted by *HB*, the score of a job is set equal to its rank. That is, the jobs are ordered in non-increasing order of their priority values, and the score sc_j of job *j* is then set equal to its rank in this sorted order. Another approach, denoted by value-biased (*VB*) stochastic sampling, was proposed by Cicirello and Smith [4]. In the *VB* approach, the score of a job is set equal to its priority value, i.e. $sc_j = I_j$.

The *HB* framework does not use the priorities of the underlying heuristic to their full potential, since it only uses those priorities to calculate the rank of each job. The *VB* approach fully utilizes the discriminatory power of the underlying dispatching heuristic, since the score of a job is set equal to its priority value. Therefore, and on the one hand, the *VB* framework distinguishes more clearly

between scenarios where the priority values are closer, or quite distant. On the other hand, the *HB* approach can provide a better discrimination among the candidate jobs when all the priority values are quite close. A more detailed analysis and comparison of these two approaches can be found in [4].

Finally, in step 2.4 a bias function is applied to the score values, in order to obtain a biased score, which is then used in the next step to calculate the selection probability for each job. The use of a bias function provides a means of placing more or less emphasis on following the advice of the underlying dispatching heuristic and its priority index. Stronger (weaker) bias functions can be used when the underlying heuristic is stronger (weaker), or when it provides values in a very small (wider) range. For a detailed discussion of bias functions and their selection, please see [3, 4].

Bresina [3] proposed several bias functions, including polynomial, exponential and random bias functions, while Cicirello and Smith [4] used the polynomial bias function. In this paper, we consider only exponential and random bias functions.

In the exponential bias function, the biased score of job j bsc_j is set equal to $exp\_base^{-sc_j}$ and $exp\_base^{sc_j}$ when the *HB* and *VB* strategies are used, respectively, where *exp_base* is the user-defined base of the exponential expression.

The polynomial bias function (as well as several of the other functions proposed in [3]) was not considered, since it is not suited to negative priority values. Indeed, the polynomial bias function requires a non-negative score. This was the case for the ranks used in the *HB* approach of Bresina [3], and for the priority values of the underlying heuristic considered in Cicirello and Smith [4]. However, the priority index of the underlying dispatching heuristic used for the considered problem can generate negative priority values. Therefore, the exponential bias function was chosen since this was the only bias function (in addition to the random bias function) that could handle the negative score values that can appear when the *VB* strategy is used.

In the random bias function, the biased score bsc_j is set equal to 1, for all jobs $j \in CL$. Therefore, all the jobs in the candidate list have an equal probability of being selected. This bias function is only used when the *RCL* strategy is employed in step 2.2. In this case, we are selecting at random from a restricted set that contains the jobs with the best priority values. The random bias function is not used in

conjunction with the *All* approach, since that would correspond to generating completely random sequences.

The proposed heuristics

In this section, we describe the heuristics that were considered. A total of ten heuristics were analysed. Five heuristics were first obtained by using different combinations of the possible strategies described in the previous section for steps 2.2, 2.3 and 2.4 of Procedure 1. The other five versions are then quite similar to these heuristics, and result simply from using a local search procedure after each greedy randomized construction, instead of only at the end of all greedy randomized constructions. These heuristics are described in detail in the next two subsections.

Heuristics with a final improvement step

The pseudo-code for the first five heuristics that were considered is given in Algorithm 1. These heuristics share the same framework, and differ only in the strategies chosen for steps 2.2, 2.3 and 2.4 of Procedure 1.

Algorithm 1: Greedy Randomized Dispatching Rule with Improvement Step

1. Generate the ETP sequence and calculate its objective function value (ofv). Set S_{best} and ofv_{best} to the ETP sequence and ofv, respectively.
2. Set $iter\_no\_improv = 0$.
3. While ($iter\_no\_improv < max\_iter\_no\_improv$)
   3.1. Apply Procedure 1 to obtain a schedule and then calculate its ofv. Set S and ofv_S equal to that schedule and its ofv, respectively.
   3.2. If $ofv_S < ofv_{best}$, set $S_{best} = S$, $ofv_{best} = ofv_S$ and $iter\_no\_improv = 0$. Otherwise, set $iter\_no\_improv = iter\_no\_improv + 1$.
4. Apply the 3SW improvement procedure to the best schedule found S_{best}. Set S_{best} and ofv_{best} to the sequence and ofv, respectively, of the schedule obtained after the application of the improvement procedure.
5. Return the schedule in S_{best} and its ofv ofv_{best}.

The underlying dispatching rule (and its corresponding priority index) is the ETP_v2 heuristic presented in Valente and Alves [25]. Indeed, this dispatching

rule provided the best performance among all the dispatching heuristics analysed in [25]. In the following, the ETP_v2 dispatching rule will be denoted simply as ETP. In step 1, we first apply the ETP rule without any greedy randomization. Just as in Cicirrelo and Smith [4], this is done in order to assure that the algorithm will always generate the schedule that would be obtained by applying the underlying heuristic.

In step 3, the algorithm uses Procedure 1 to generate greedy randomized sequences, updating the best schedule found (S_{best}) and its objective function value (ofv_{best}) when appropriate. This is repeated until the stopping criterion is met. For the stopping criterion, we have selected the number of iterations without improvement in the best solution found. Therefore, the algorithm stops the greedy randomized constructions when the number of consecutive iterations without improvement in the best schedule found (*iter_no_improv*) reaches the maximum user-defined value *max_iter_no_improv*.

In step 4, an improvement step is applied to the best schedule found. The 3-swap (3SW) improvement procedure was selected, since it was recommended among all the simple improvement procedures analysed in Valente and Alves [25]. The 3SW method was also applied, as an improvement step, to the beam search heuristics considered in [21].

As previously mentioned, this common framework is shared by the first five heuristics, that then differ only in the choices made for steps 2.2, 2.3 and 2.4 in Procedure 1. In the heuristic denoted by RCL, the *RCL* strategy and the random bias function are used in steps 2.2 and 2.4, respectively. Since the random bias function simply sets the biased score of each candidate job equal to 1, step 2.3 can be skipped. The heuristic All_HB (All_VB) uses the *All* strategy, the *HB* (*VB*) approach and the exponential bias function in steps 2.2, 2.3 and 2.4, respectively. Finally, the RCL_HB (RCL_VB) heuristic is similar to the All_HB (All_VB) procedure, with the exception that the *RCL* strategy is used instead of the *All* approach in step 2.2.

Heuristics with local search

The pseudo-code for the remaining five heuristics that were considered is given in Algorithm 2.

Algorithm 2: Greedy Randomized Dispatching Rule with Local Search

1. Generate the ETP sequence and apply the 3SW improvement procedure to that sequence. Set S_{best} and ofv_{best} to the sequence obtained after the improvement procedure and its ofv, respectively.
2. Set $iter\_no\_improv = 0$.
3. While ($iter\_no\_improv < max\_iter\_no\_improv$)
   3.1. Apply Procedure 1 to obtain a schedule, and then apply the 3SW improvement procedure to that schedule. Set S and ofv_S equal to the schedule obtained after the improvement procedure and its ofv, respectively.
   3.2. If $ofv_S < ofv_{best}$, set $S_{best} = S$, $ofv_{best} = ofv_S$ and $iter\_no\_improv = 0$. Otherwise, set $iter\_no\_improv = iter\_no\_improv + 1$.
4. Return the schedule in S_{best} and its ofv ofv_{best}.

These five heuristics use the same combinations of strategies for steps 2.2, 2.3 and 2.4 in Procedure 1 as the heuristics described in the previous subsection. Therefore, and as can be seen from the pseudo-code of Algorithm 2, these versions differ from the previous ones only in the fact that the 3SW improvement procedure is applied to all the constructed schedules, instead of being applied only to the best of the constructed schedules, as a final improvement step. These procedures will be denoted by appending "_3SW" to the identifiers of the heuristics described in the previous subsection. These heuristics can be seen as simple GRASP algorithms, since each of the schedule constructions is followed by a local search procedure.

We remark that the heuristics described in the previous subsection do not guarantee a better or equal schedule than the ETP dispatching rule (the subjacent heuristic) followed by the 3SW improvement procedure. The schedule obtained before the improvement step cannot be worse than the ETP schedule, since this schedule is generated in step 1 of Algorithm 1. However, due to the application of the final improvement step, it cannot be guaranteed that the procedures described in the previous subsection are not outperformed by the ETP dispatching rule followed by the 3SW improvement procedure.

The _3SW versions presented in this section, on the other hand, are guaranteed to generate a final schedule that is at least as good as that of the ETP rule plus the

3SW improvement procedure. Indeed, that schedule is generated in step 1 of these _3SW versions.

Computational results

In this section, we first present the set of test problems used in the computational tests. Then, the preliminary experiments that were performed to determine appropriate values for the parameters required by the heuristics are described. Finally, the computational results are presented. We first compare the greedy randomized constructive heuristics with the existing procedures, 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, 750 and 1000 jobs. These problems were randomly generated as follows. For each job 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, 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 RCL_HB, RCL_HB_3SW, RCL_VB and RCL_VB_3SW procedures were not applied to the 1000 job instances, and the All_HB, All_HB_3SW, All_VB and All_VB_3SW heuristics were only used on instances with up to 500 jobs.

Preliminary tests

Extensive preliminary test were conducted in order to determine adequate values for the parameters required by the several heuristics. A separate problem set was used to perform these preliminary experiments. This test set included instances with 10, 25, 50, 100, 250, 500, 1000 and 2000 instances, and contained 5 instances for each combination of problem size, processing time and penalty variability, T and R. The instances in this smaller test set were generated just as previously described for the full problem set.

The RCL heuristic requires a value for the threshold parameter α. An initial range of values between 0 and 1 was first considered, and the objective function value was computed for each value of α and each instance. These results were then analysed, and several successive additional ranges were then tested, with the values for each successive range being chosen according to the results obtained for the previous ranges. Based on these tests, the following formula was then chosen for setting the value of the threshold parameter: $\alpha = 0.1 / n$.

A similar approach was also used to determine an adequate value for the *exp_base* parameter required by the All_HB and All_VB heuristics. A fixed value of 4 proved appropriate for the All_HB procedure. For the All_VB heuristic, however, the formula $1 + 0.1n^{-0.33}$ was chosen for the *exp_base* parameter.

The RCL_HB and RCL_VB heuristics require values for both the α and *exp_base* parameters. For these two heuristics, the following strategy was used to determine appropriate values for these parameters. On the one hand, we first set the threshold parameter at the formula previously derived for the RCL heuristic. Then, experiments similar to those described before were performed to determine an adequate value for the *exp_base* parameter. On the other hand, we also tried the opposite approach. That is, the *exp_base* parameter was first set at the value / formula previously obtained for the All_HB and All_VB heuristics, and experiments were then conducted to determine an appropriate value for the threshold parameter α.

For both the RCL_HB and RCL_VB heuristics, the second approach provided better results. Therefore, the *exp_base* parameter was set at 4 and $1 + 0.1n^{-0.33}$ for the RCL_HB and RCL_VB algorithms, respectively. The threshold parameter α was then set at 0.05 for $n \leq 50$ and 0.002 when $n > 50$ for the RCL_HB heuristic.

For the RCL_VB procedure, this parameter was instead set at 0.5 for $n \leq 25$, 0.05 when $25 < n < 100$ and 0.002 when $n \geq 100$.

The five _3SW heuristics also require values for the threshold parameter α and / or the *exp_base* parameter. For each of these heuristics, these parameters were set equal to the values / formulas previously determined for their non _3SW counterparts, after some experiments showed that this approach indeed provided adequate results.

The proposed heuristics also require a value for the maximum number of consecutive iterations without improvement in the best schedule found *max_iter_no_improv*. Since larger values of this parameter are likely to be required for larger instances, we considered several formulas that calculate the *max_iter_no_improv* parameter as an increasing function of the instance size *n*. More specifically, for the non _3SW versions the three function $5n^{0.305}$, $3.7n^{0.435}$ and $17.6n^{0.23}$ were tested. For these functions, and as *n* increases from 10 to 2000, the value of *max_iter_no_improv* ranges from 10 to 50, 10 to 100 and 30 to 100, respectively.

The _3SW versions are likely to require a lower value for the parameter *max_iter_no_improv*, since they apply an improvement procedure to each of the constructed solutions. Therefore, for the _3SW versions we considered the functions $2.5n^{0.305}$, $1.85n^{0.435}$ and $5n^{0.305}$. For these functions, the value of *max_iter_no_improv* ranges from 5 to 25, 5 to 50 and 10 to 50, respectively, as *n* goes from 10 up to 2000. The objective function values were then obtained for each instance and each of the functions for the parameter *max_iter_no_improv*. These results were analysed, and the functions $17.6n^{0.23}$ and $5n^{0.305}$ were then selected for the non _3SW and the _3SW versions, respectively.

Finally, we remark that even though instances with up to 2000 jobs were used in the preliminary tests, the computational results presented in the next subsections were then obtained on a set that included instances with only up to 1000 jobs, as previously mentioned. This was necessary in order to keep the total computational time within acceptable limits. In fact, and on the one hand, the full test set is much larger than the set used in the preliminary tests. On the other hand, as described in the next subsection, the heuristics were also applied not one but several times to each instance, with different random number seeds, in order to best ascertain their average behaviour.

Comparison with existing heuristics

In this subsection, the proposed greedy randomized dispatching rules are compared with existing heuristics for the considered problem. More specifically, the greedy randomized heuristics are compared with the ETP dispatching rule presented in [25] and the recovering beam search algorithm, denoted by RBS, proposed in [21]. Both these algorithms include a final improvement step that uses the 3SW improvement method. The ETP rule, as previously mentioned, provided the best results among the dispatching heuristics analysed in [25]. The RBS procedure, on the other hand, is currently the best performing heuristic for the single machine quadratic earliness and tardiness problem. For each instance, 10 independent runs, with different random number seeds, were performed for all the greedy randomized dispatching rules.

Table 1 provides the mean relative improvement in objective function value over the ETP and the RBS procedures (denoted by %imp – ETP and %imp – RBS, respectively). In table 2, we give the percentage number of times four selected greedy randomized heuristic versions perform better (<), equal (=) or worse (>) than the ETP and RBS procedures.

The relative improvement over the ETP and RBS heuristics is calculated as (*heur_ofv – grdr_ofv*) / *heur_ofv* * 100, where *heur_ofv* and *grdr_ofv* are the objective function values of the appropriate heuristic (ETP or RBS) and the appropriate greedy randomized dispatching rule, respectively. The avg (best) column provides the relative improvement calculated with the average (best) of the objective function values obtained for all the 10 runs. The results in the avg column provide an indication of the relative improvement we can achieve if the procedure is executed only once, while the best column shows the improvement that can be obtained if the algorithm is allowed to perform 10 runs.

The processing time and penalty variability has a significant impact on the difficulty of the problem, and therefore on the results obtained by the several heuristic procedures and their comparison. When the variability is low, the problem is much easier, and even simple procedures obtain optimum or near optimum results, so there is little or no room for improvement. This has been previously established and discussed in [25, 21], and will also be shown quite clearly by the comparison with the optimum results provided in the next subsection.

For the low variability instances, the performance of the several procedures is virtually identical. Even though there are differences in the objective function values, as evidenced by the results in table 2, these values are nevertheless extremely close, as can be seen by the 0.00 relative improvements given in table 1. When the variability is high, however, the problem becomes significantly more difficult, and the difference in performance between the several algorithms is much more noticeable.

The _3SW versions are clearly superior to their non _3SW counterparts, as can be seen by the relative improvement values and the objective function comparison provided in tables 1 and 2. This is to be expected, since the _3SW versions apply an improvement procedure to each constructed schedule, instead of only applying it to the best of the generated schedules, as a final improvement step.

The All_VB and RCL_VB are the best of the greedy randomized dispatching rules. For the small to medium instances, the All_HB and RCL_HB outperform the RCL heuristic. However, for the medium to large instances, the RCL algorithm is superior to the _HB heuristics.

The All_VB (All_HB) and the RCL_VB (RCL_HB) provide similar results in terms of solution quality, particularly for the larger instances, where the performance of these heuristics is virtually identical. The *VB* approach is clearly superior, for the considered problem, to the *HB* strategy. Therefore, and in this particular case, the more intensive use of the discriminatory power of the underlying heuristic allowed by the *VB* approach leads to a better performance.

All of the proposed heuristics clearly outperform the ETP dispatching rule (their underlying heuristic). Therefore, the greedy randomization of this dispatching heuristic indeed proved to be an effective way of improving its performance for the considered problem. The relative improvement over the ETP rule, although always clearly positive, tends to decrease with the instance size.

When the best result over the 10 runs are considered, the best performing of the proposed heuristic (All_VB_3SW, RCL_VB_3SW and their non _3SW counterparts) provide a minor positive relative improvement over the RBS algorithm. However, when the average result is used, the relative improvement is negative. Therefore, when the average performance is considered, the RBS procedure outperforms the proposed heuristics, particularly in small to medium size instances.

Indeed, as the instance size increases, and as can be seen by the results in tables 1 and 2, the average performance of the proposed heuristics versus the RBS algorithm improves. For the largest instances, as seen in table 1, the average results of the best performing of the proposed heuristics are quite close, or nearly identical, to those of the RBS procedure. This is particularly relevant, since dispatching rules and greedy randomized dispatching rules are usually heuristics that are suited to large instances that cannot be solved in reasonable time by other types of heuristic procedures.

Therefore, it is actually quite positive that, as the instance size increases, the performance of the greedy randomized heuristics becomes similar to that of the best existing heuristic for small to medium size instances. Indeed, this seems to indicate that the level of performance allowed by the RBS heuristic for small to medium size instances can also be achieved by the best of the proposed greedy randomized heuristics for large instances.

In table 3, we present the effect of the T and R parameters on the relative improvement (calculated with the average objective function value) over the ETP dispatching rule, for instances with 100 jobs. The relative improvement is quite minor for the extreme values of T ($T = 0.0$ and $T = 1.0$) When the tardiness factor assumes more intermediate values, the relative difference in objective function values becomes larger.

This is in accordance with the results previously obtained in [25, 21]. Indeed, the problem is much easier when most jobs are early ($T = 0.0$) or tardy ($T = 1.0$); again, this will also be shown quite clearly in the next subsection. For the more intermediate values of T, the number of early and tardy jobs becomes more balanced, and the problem becomes harder. Therefore, there is more room for improvement in the harder instances with intermediate values of the tardiness factor.

The heuristic runtimes (in seconds) are presented in table 4. For the proposed heuristics, we give the average runtime, i.e. the average of the runtimes for each of the 10 runs. The RBS algorithm is computationally demanding, and can only be applied to small and medium size instances. As expected, the ETP dispatching rule is quite clearly the most efficient of the heuristic procedures. The proposed heuristics, though naturally more demanding than the ETP rule, are still

computationally efficient, and can solve even large instances in reasonable computation times.

The _3SW versions are faster than their non _3SW counterparts, which may seem surprising. However, this can be explained by the stopping criterion that was chosen, i.e. the number of iterations without improving the best solution found. The _3SW versions apply the improvement procedure to each constructed solution, and can therefore reach very good solutions in few iterations. The non _3SW versions only apply the improvement procedure at the end of the constructions, and may then require many more constructions / iterations to generate very good schedules. This is evidenced by the fact that, as mentioned in the previous subsection, the preliminary tests showed that a lower maximum number of iterations without improvement could be used in the _3SW versions. The versions that use a *RCL* strategy are also much faster than the heuristics that include all the unscheduled jobs in the candidate list.

The RBS procedure is a good choice for small and up to medium size instances, since it can solve these instances within reasonable computation times, and its performance is better than the average performance of the proposed procedures. For larger instances, however, the RCL_VB_3SW is the recommended heuristic. Actually, this procedure is also a good alternative to the RBS algorithm for the largest of the middle size instances, since its average performance is similar to that of the RBS procedure for those instances.

The RCL_VB_3SW procedure is also the best performing of the proposed heuristics, along with the All_VB_3SW algorithm, in terms of solution quality. However, the RCL_VB_3SW heuristic is much more computationally efficient that its All_ counterpart.

Comparison with optimum results

In this subsection, the heuristics are compared with optimum objective function values, for instances with up to 20 jobs. Table 5 gives 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 provided.

The results given in table 5 confirm that, as mentioned in the previous subsection, the problem becomes significantly more difficult when the processing time and penalty variability is high. For the low variability instances, all the heuristic procedures, including the simple ETP dispatching rule, provide optimum results for nearly all instances. When the variability is high, however, the problem becomes harder, and there is more room to improve upon the ETP results. Indeed, all the other heuristic procedures clearly outperform the ETP rule for the high variability instances.

The RBS algorithm performs quite well, both providing a low relative deviation from the optimum, and obtaining the optimum solution for a large percentage of the instances. The best of the proposed heuristics also perform quite well. In fact, their relative deviation from the optimum is usually less than 1%, and they achieve an optimum solution for over 80% of the instances. The RCL and RCL_3SW are the worst performing of the proposed heuristics on these smallest instances. Nevertheless, they are still only about 3% above the optimum, and usually obtain optimum results for over 70% of the instances.

In table 6, we present the effect of the T and R parameters on the relative deviation from the optimum, for instances with 20 jobs. Again, the results given in this table confirm that, as previously mentioned, the problem is much harder when there is a greater balance between the number of early and tardy jobs.

In fact, when $T \leq 0.2$ or $T \geq 0.8$, all the heuristics are optimal or nearly optimal. However, the relative deviation from the optimum is higher when $T = 0.4$ and $T = 0.6$, particularly when the due date range is low. For these instances, the relative improvement over the ETP rule given by the RBS algorithm and the proposed heuristic procedures is much higher.

Conclusion

In this paper, we presented greedy randomized dispatching heuristics for the single machine scheduling problem with quadratic earliness and tardiness costs, and no machine idle time. Several different heuristic versions were considered. On the one hand, these versions correspond to different combinations of strategies for the construction of the candidate list, the calculation of the biased score and the choice of bias function. On the other hand, we also considered versions with

only a final improvement step, as well as versions where a local search is performed after each greedy randomized construction.

Extensive initial experiments were first performed, in order to determine adequate values for the parameters required by the proposed heuristics. These heuristics were then compared with the ETP dispatching rule (their underlying heuristic) and the RBS algorithm (the best of the existing heuristic procedures), as well as with optimum solutions for the smaller instance sizes.

The _3SW versions were superior to their non _3SW counterparts, in both solution quality and computation time. The All_ and RCL_ heuristics provided similar results in terms of solution quality, but the RCL_ versions were significantly faster. Also, the *VB* approach was clearly superior to the *HB* strategy.

The proposed heuristic procedures noticeably outperformed the ETP rule (their underlying heuristic), so the greedy randomization of this dispatching rule was indeed an effective way of improving its performance. The best of the proposed heuristics also provided results that were quite close to the optimum, and generated an optimum solution for over 80% of the instances.

The RBS procedure is still a good choice for small and medium size instances. Indeed, this algorithm not only can solve these instances within reasonable computation times, but its performance is also usually better than the average performance of the proposed procedures.

For larger instances, however, the RBS algorithm is too computationally demanding, and the RCL_VB_3SW procedure is then the heuristic of choice. Actually, this procedure is also an alternative to the RBS algorithm for the larger of the middle size instances, since for these instances its average performance is similar to that of the RBS heuristic. Therefore, the best of the proposed algorithms is an alternative for medium size instances, and the new recommended heuristic for the large instances.

References

1. Abdul-Razaq T, Potts CN (1988) Dynamic programming state-space relaxation for single machine scheduling. J Oper Res Soc 39:141-152.

2. Baker KR, Scudder GD (1990) Sequencing with earliness and tardiness penalties: A review. Oper Res 38:22-36.

3. Bresina JL (1996) Heuristic-biased stochastic sampling. In: Proceedings of the Tirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference, Volume One, AAAI Press, pp 271-278.
4. Cicirello VA, Smith SF (2005) Enhancing stochastic search performance by value-biased randomization of heuristics. J Heuristics 11:5-34.
5. Gupta SK, Sen T (1983) Minimizing a quadratic function of job lateness on a single machine. Eng Costs Prod Econ 7:187-194.
6. Hoogeveen H (2005) Multicriteria scheduling. Eur J Oper Res 167:592-623.
7. Kanet JJ, Sridharan V (2000) Scheduling with inserted idle time: Problem taxonomy and literature review. Oper Res 48:99-110.
8. Korman K (1994). A pressing matter. Video February:46-50.
9. 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.
10. Li G (1997) Single machine earliness and tardiness scheduling. Eur J Oper Res 96:546-558.
11. Liaw CF (1999) A branch-and-bound algorithm for the single machine earliness and tardiness scheduling problem. Comp Oper Res 26:679-693.
12. Ow PS, Morton TE (1989) The single machine early/tardy problem. Manag Sci 35:177-191.
13. Resende MGC, Ribeiro CC (2003) Greedy randomized adaptive search procedures. In: Glover F, Kochenberger GA (eds) Handbook of metaheuristics. Kluwer Academic Publishers, Dordrecht, pp 219-249.
14. Schaller J (2002) Minimizing the sum of squares lateness on a single machine. Eur J Oper Res 143:64-79.
15. Schaller J (2004) Single machine scheduling with early and quadratic tardy penalties. Comp Ind Eng 46:511-532.
16. Sen T, Dileepan P, Lind MR (1995) Minimizing a weighted quadratic function of job lateness in the single machine system. Int J Prod Econ 42:237-243.
17. Su LH, Chang PC (1998) A heuristic to minimize a quadratic function of job lateness on a single machine. Int J Prod Econ 55:169-175.
18. Valente JMS (2007a) An exact approach for single machine scheduling with quadratic earliness and tardiness penalties. Working Paper 238, Faculdade de Economia, Universidade do Porto, Portugal.
19. Valente JMS (2007b) Heuristics for the single machine scheduling problem with early and quadratic tardy penalties. Eur J Ind Eng 1:431-448.
20. Valente JMS (2008a) An exact approach for the single machine scheduling problem with linear early and quadratic tardy penalties. Asia-Pac J Oper Res 25:169-186.
21. Valente JMS (2008b) Beam search heuristics for quadratic earliness and tardiness scheduling. Working Paper 279, Faculdade de Economia, Universidade do Porto, Portugal.
22. Valente JMS, Alves RAFS (2005a) Filtered and recovering beam search algorithms for the early/tardy scheduling problem with no idle time. Comp Ind Eng 48:363-375.
23. Valente JMS, Alves RAFS (2005b) Improved heuristics for the early/tardy scheduling problem with no idle time. Comp Oper Res 32:557-569.

24. Valente JMS, Alves RAFS (2005c) Improved lower bounds for the early/tardy scheduling problem with no idle time. J Oper Res Soc 56:604-612.
25. Valente JMS, Alves RAFS (2008) Heuristics for the single machine scheduling problem with quadratic earliness and tardiness penalties. Comp Oper Res 35:3696-3713.

Table 1 Comparison with the ETP and RBS heuristics – relative improvement

		low var				high var			
		%imp - ETP		%imp - RBS		%imp - ETP		%imp - RBS	
n	heur	best	avg	best	avg	best	avg	best	avg
50	RCL	0.00	0.00	0.00	0.00	1.46	1.22	-0.76	-1.03
	All_HB	0.00	0.00	0.00	0.00	2.18	1.39	0.06	-0.81
	All_VB	0.00	0.00	0.00	0.00	2.45	1.88	0.36	-0.27
	RCL_HB	0.00	0.00	0.00	0.00	2.24	1.51	0.12	-0.69
	RCL_VB	0.00	0.00	0.00	0.00	2.47	1.89	0.38	-0.26
	RCL_3SW	0.00	0.00	0.00	0.00	1.68	1.41	-0.53	-0.83
	All_HB_3SW	0.00	0.00	0.00	0.00	2.33	1.58	0.22	-0.63
	All_VB_3SW	0.00	0.00	0.00	0.00	2.58	1.89	0.49	-0.27
	RCL_HB_3SW	0.00	0.00	0.00	0.00	2.29	1.53	0.18	-0.68
	RCL_VB_3SW	0.00	0.00	0.00	0.00	2.44	1.88	0.33	-0.28
100	RCL	0.00	0.00	0.00	0.00	1.25	1.01	-0.17	-0.42
	All_HB	0.00	0.00	0.00	0.00	1.32	0.88	-0.09	-0.55
	All_VB	0.00	0.00	0.00	0.00	1.73	1.27	0.35	-0.14
	RCL_HB	0.00	0.00	0.00	0.00	1.18	0.83	-0.24	-0.61
	RCL_VB	0.00	0.00	0.00	0.00	1.48	1.13	0.08	-0.30
	RCL_3SW	0.00	0.00	0.00	0.00	1.39	1.14	-0.02	-0.29
	All_HB_3SW	0.00	0.00	0.00	0.00	1.51	1.03	0.10	-0.41
	All_VB_3SW	0.00	0.00	0.00	0.00	1.80	1.36	0.41	-0.06
	RCL_HB_3SW	0.00	0.00	0.00	0.00	1.25	0.85	-0.17	-0.59
	RCL_VB_3SW	0.00	0.00	0.00	0.00	1.51	1.16	0.11	-0.27
250	RCL	0.00	0.00	0.00	0.00	0.72	0.52	0.04	-0.17
	All_HB	0.00	0.00	0.00	0.00	0.60	0.41	-0.09	-0.28
	All_VB	0.00	0.00	0.00	0.00	0.84	0.60	0.16	-0.08
	RCL_HB	0.00	0.00	0.00	0.00	0.59	0.41	-0.09	-0.28
	RCL_VB	0.00	0.00	0.00	0.00	0.82	0.60	0.15	-0.08
	RCL_3SW	0.00	0.00	0.00	0.00	0.78	0.59	0.10	-0.10
	All_HB_3SW	0.00	0.00	0.00	0.00	0.62	0.45	-0.06	-0.24
	All_VB_3SW	0.00	0.00	0.00	0.00	0.86	0.64	0.19	-0.04
	RCL_HB_3SW	0.00	0.00	0.00	0.00	0.62	0.44	-0.07	-0.25
	RCL_VB_3SW	0.00	0.00	0.00	0.00	0.86	0.64	0.18	-0.04
500	RCL	0.00	0.00	0.00	0.00	0.43	0.30	0.03	-0.10
	All_HB	0.00	0.00	0.00	0.00	0.31	0.22	-0.09	-0.18
	All_VB	0.00	0.00	0.00	0.00	0.50	0.36	0.11	-0.04
	RCL_HB	0.00	0.00	0.00	0.00	0.31	0.22	-0.09	-0.18
	RCL_VB	0.00	0.00	0.00	0.00	0.50	0.36	0.10	-0.04
	RCL_3SW	0.00	0.00	0.00	0.00	0.46	0.35	0.07	-0.05
	All_HB_3SW	0.00	0.00	0.00	0.00	0.33	0.25	-0.07	-0.15
	All_VB_3SW	0.00	0.00	0.00	0.00	0.52	0.40	0.13	0.00
	RCL_HB_3SW	0.00	0.00	0.00	0.00	0.33	0.24	-0.07	-0.15
	RCL_VB_3SW	0.00	0.00	0.00	0.00	0.53	0.40	0.14	0.00

Table 2 Comparison with the ETP and RBS heuristics – percentage of better, equal and worse results

			All_VB			RCL_VB			All_VB_3SW			RCL_VB_3SW		
heur	var	n	<	=	>	<	=	>	<	=	>	<	=	>
ETP	L	25	0.1	99.9	0.0	0.2	99.8	0.0	3.4	96.6	0.0	3.3	96.7	0.0
		50	0.0	100.0	0.0	0.4	99.6	0.0	8.2	91.8	0.0	7.1	92.9	0.0
		75	0.0	100.0	0.0	0.1	99.9	0.0	12.0	88.0	0.0	11.0	89.0	0.0
		100	0.0	100.0	0.0	1.5	97.8	0.7	15.0	85.0	0.0	2.3	97.7	0.0
		250	0.0	100.0	0.0	2.9	96.5	0.6	31.1	68.9	0.0	18.5	81.5	0.0
		500	0.0	100.0	0.0	0.6	99.1	0.3	47.8	52.2	0.0	41.5	58.5	0.0
	H	25	27.6	70.4	2.0	27.5	70.5	2.0	33.0	67.0	0.0	32.9	67.1	0.0
		50	31.9	65.2	3.0	33.3	63.4	3.2	49.4	50.7	0.0	48.3	51.7	0.0
		75	30.0	67.8	2.2	32.5	64.9	2.6	59.9	40.1	0.0	59.7	40.4	0.0
		100	31.1	67.2	1.7	33.9	63.6	2.6	64.1	36.0	0.0	55.6	44.4	0.0
		250	26.9	71.9	1.2	31.3	66.1	2.6	79.6	20.4	0.0	75.8	24.2	0.0
		500	26.4	72.5	1.2	27.7	71.0	1.3	86.1	13.9	0.0	86.2	13.8	0.0
RBS	L	25	0.3	96.1	3.6	0.3	96.1	3.6	0.6	98.9	0.6	0.6	98.8	0.6
		50	0.7	92.4	6.9	0.8	92.5	6.7	2.6	96.7	0.7	2.4	95.9	1.7
		75	1.3	91.2	7.5	1.3	91.3	7.4	6.5	92.7	0.8	6.1	92.1	1.8
		100	1.8	88.1	10.2	2.0	88.2	9.8	9.0	89.8	1.2	2.7	88.3	9.1
		250	4.0	83.0	13.0	4.7	82.6	12.7	25.7	72.3	2.0	15.0	76.5	8.6
		500	7.0	76.5	16.5	7.0	76.5	16.5	42.4	55.7	1.9	35.0	58.2	6.8
	H	25	7.8	72.9	19.3	7.6	73.0	19.4	10.9	75.6	13.6	10.8	75.7	13.5
		50	16.0	52.7	31.2	16.3	53.2	30.6	24.0	57.6	18.4	23.0	57.6	19.4
		75	19.5	43.5	37.0	20.2	43.7	36.1	34.9	42.8	22.3	34.4	43.2	22.4
		100	21.0	42.5	36.5	21.2	41.0	37.8	41.5	34.8	23.8	31.8	40.2	28.0
		250	17.1	47.4	35.4	20.8	44.1	35.1	59.0	17.6	23.4	55.1	21.6	23.3
		500	17.3	48.1	34.6	18.1	47.6	34.4	66.8	10.9	22.3	66.6	11.3	22.2

Table 3 Relative improvement over the ETP heuristic for instances with 100 jobs

		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
All_VB	0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0005	0.0000
	0.2	0.0000	0.0000	0.0000	0.0000	0.1295	0.0370	0.0297	0.0034
	0.4	0.0000	0.0000	0.0000	0.0000	1.9627	1.4985	1.2954	0.7138
	0.6	0.0000	0.0000	0.0000	0.0000	5.6089	5.1189	5.7313	2.3034
	0.8	0.0000	0.0000	0.0000	0.0000	5.5368	0.5944	0.0212	0.0093
	1.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0005	0.0003	0.0007
RCL_VB	0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0005	0.0000
	0.2	0.0000	0.0000	0.0000	0.0000	0.1006	0.0338	0.0300	0.0037
	0.4	0.0000	0.0000	0.0001	0.0001	1.5888	1.3444	1.1826	0.6494
	0.6	0.0000	0.0000	0.0000	0.0001	4.4836	4.4677	5.2286	2.3719
	0.8	0.0000	0.0000	0.0000	0.0000	5.1176	0.4817	0.0186	0.0036
	1.0	-0.0003	0.0000	0.0000	0.0000	0.0000	0.0005	0.0000	0.0003
All_VB_3SW	0.0	0.0002	0.0000	0.0000	0.0000	0.0002	0.0007	0.0009	0.0004
	0.2	0.0020	0.0000	0.0000	0.0000	0.1569	0.0479	0.0376	0.0074
	0.4	0.0018	0.0000	0.0001	0.0001	1.9915	1.7606	1.5165	1.0325
	0.6	0.0021	0.0001	0.0000	0.0001	5.4508	5.3918	6.2444	3.2295
	0.8	0.0014	0.0000	0.0000	0.0000	5.0647	0.6185	0.0173	0.0158
	1.0	0.0001	0.0001	0.0000	0.0000	0.0005	0.0011	0.0007	0.0012
RCL_VB_3SW	0.0	0.0000	0.0000	0.0000	0.0000	0.0003	0.0007	0.0009	0.0004
	0.2	0.0000	0.0000	0.0000	0.0000	0.1094	0.0433	0.0386	0.0069
	0.4	0.0000	0.0000	0.0001	0.0001	1.5974	1.5270	1.2265	0.9139
	0.6	0.0000	0.0000	0.0000	0.0001	4.2658	4.5386	5.4324	2.9087
	0.8	0.0000	0.0000	0.0000	0.0000	4.7235	0.5193	0.0119	0.0044
	1.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0006	0.0000	0.0003

Table 4 Runtimes (in seconds)

var	heur	n=25	n=50	n=75	n=100	n=250	n=500	n=750
L	ETP	0.0002	0.0004	0.0004	0.0009	0.0037	0.0129	0.0273
	RBS	0.0071	0.0366	0.1043	0.2263	3.2328	25.8692	---
	RCL	0.0017	0.0069	0.0174	0.0341	0.3011	1.2955	3.0234
	All_HB	0.0043	0.0198	0.0474	0.0911	0.7134	3.5024	---
	All_VB	0.0045	0.0197	0.0471	0.0877	0.6636	3.2884	---
	RCL_HB	0.0023	0.0074	0.0191	0.0386	0.2118	0.8596	2.0613
	RCL_VB	0.0031	0.0073	0.0164	0.0376	0.1948	0.8424	2.0326
	RCL_3SW	0.0021	0.0060	0.0118	0.0196	0.1072	0.4557	1.0853
	All_HB_3SW	0.0039	0.0136	0.0293	0.0535	0.3810	1.9825	---
	All_VB_3SW	0.0041	0.0148	0.0320	0.0581	0.3978	2.0601	---
	RCL_HB_3SW	0.0024	0.0075	0.0119	0.0199	0.1144	0.5246	1.3270
	RCL_VB_3SW	0.0034	0.0082	0.0169	0.0201	0.1174	0.5460	1.3843
H	ETP	0.0002	0.0004	0.0007	0.0010	0.0042	0.0137	0.0285
	RBS	0.0073	0.0382	0.1087	0.2400	3.3756	27.5335	---
	RCL	0.0021	0.0081	0.0183	0.0337	0.2447	1.1170	2.7233
	All_HB	0.0052	0.0231	0.0557	0.1076	0.8594	4.3028	---
	All_VB	0.0072	0.0296	0.0667	0.1247	1.0064	5.3147	---
	RCL_HB	0.0035	0.0149	0.0213	0.0379	0.2693	1.2518	3.1798
	RCL_VB	0.0057	0.0169	0.0382	0.0395	0.2723	1.2448	3.1108
	RCL_3SW	0.0031	0.0098	0.0202	0.0346	0.2070	0.8798	2.0231
	All_HB_3SW	0.0049	0.0179	0.0404	0.0753	0.5544	2.8535	---
	All_VB_3SW	0.0045	0.0177	0.0414	0.0788	0.6374	3.5226	---
	RCL_HB_3SW	0.0037	0.0133	0.0181	0.0323	0.2106	0.9951	2.4595
	RCL_VB_3SW	0.0042	0.0131	0.0298	0.0340	0.2223	1.0114	2.4695

Table 5 Comparison with optimum objective function values

		n=10		n=15		n=20	
var	heur	%dev	%opt	%dev	%opt	%dev	%opt
L	ETP	0.01	98.50	0.00	97.92	0.00	96.58
	RBS	0.00	99.92	0.00	100.00	0.00	99.67
	RCL	0.01	98.50	0.00	98.08	0.00	96.75
	All_HB	0.00	98.82	0.00	98.33	0.00	96.94
	All_VB	0.00	98.94	0.00	98.18	0.00	96.74
	RCL_HB	0.01	98.53	0.00	98.74	0.00	97.93
	RCL_VB	0.00	99.06	0.00	98.18	0.00	96.83
	RCL_3SW	0.01	98.50	0.00	98.16	0.00	96.83
	All_HB_3SW	0.00	99.37	0.00	99.19	0.00	98.72
	All_VB_3SW	0.00	99.82	0.00	99.80	0.00	99.73
	RCL_HB_3SW	0.01	98.50	0.00	98.78	0.00	97.94
	RCL_VB_3SW	0.00	99.65	0.00	99.75	0.00	99.64
H	ETP	4.69	80.75	5.17	70.67	5.89	64.83
	RBS	0.22	95.67	0.91	87.92	1.40	82.50
	RCL	2.98	84.67	2.97	74.66	3.86	68.32
	All_HB	0.30	89.19	0.62	77.13	1.11	69.43
	All_VB	0.31	91.75	0.65	81.39	0.96	74.97
	RCL_HB	1.22	88.24	0.94	78.08	1.11	71.86
	RCL_VB	0.27	91.66	0.66	81.11	0.90	75.52
	RCL_3SW	2.89	86.38	2.70	77.88	3.43	73.60
	All_HB_3SW	0.09	93.78	0.47	85.94	0.92	80.49
	All_VB_3SW	0.31	93.22	0.63	85.63	1.03	80.48
	RCL_HB_3SW	0.97	89.74	0.85	82.07	1.22	77.60
	RCL_VB_3SW	0.29	93.12	0.71	85.74	1.03	80.36

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

		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	0.0004	0.0000	0.0000	0.0000	0.0054	0.0543	0.0140	0.0001
	0.2	0.0317	0.0002	0.0000	0.0000	0.5377	0.2511	0.2774	0.0913
	0.4	0.0029	0.0000	0.0000	0.0000	13.3533	14.8223	9.4076	10.5652
	0.6	0.0112	0.0000	0.0033	0.0000	33.8728	20.3533	12.3946	10.0530
	0.8	0.0030	0.0002	0.0002	0.0005	10.2736	3.7647	0.8524	0.3565
	1.0	0.0013	0.0002	0.0000	0.0000	0.0396	0.0298	0.0300	0.0112
RBS	0.0	0.0001	0.0000	0.0000	0.0000	0.0000	0.0003	0.0005	0.0000
	0.2	0.0000	0.0000	0.0000	0.0000	0.1049	0.0038	0.0000	0.0049
	0.4	0.0000	0.0000	0.0000	0.0000	3.8803	6.2191	0.6814	0.8040
	0.6	0.0000	0.0004	0.0000	0.0000	10.9426	3.6340	2.6383	0.7352
	0.8	0.0000	0.0000	0.0000	0.0000	2.6138	1.0529	0.0310	0.1715
	1.0	0.0005	0.0000	0.0000	0.0000	0.0050	0.0021	0.0057	0.0000
All_VB	0.0	0.0004	0.0000	0.0000	0.0000	0.0000	0.0008	0.0005	0.0000
	0.2	0.0094	0.0002	0.0000	0.0000	0.1171	0.0351	0.0132	0.0165
	0.4	0.0029	0.0000	0.0000	0.0000	2.4528	2.7568	1.9141	1.7032
	0.6	0.0034	0.0000	0.0000	0.0000	7.4985	3.1782	2.3329	0.2891
	0.8	0.0030	0.0002	0.0002	0.0000	0.2304	0.3869	0.0000	0.0000
	1.0	0.0013	0.0002	0.0000	0.0000	0.0000	0.0000	0.0056	0.0000
RCL_VB	0.0	0.0004	0.0000	0.0000	0.0000	0.0000	0.0008	0.0005	0.0039
	0.2	0.0030	0.0002	0.0000	0.0000	0.1205	0.0351	0.0132	0.0165
	0.4	0.0029	0.0000	0.0000	0.0000	2.4932	2.9536	1.7284	2.0018
	0.6	0.0034	0.0000	0.0000	0.0000	6.5415	2.5218	2.4602	0.2088
	0.8	0.0030	0.0002	0.0002	0.0005	0.2835	0.2996	0.0001	0.0000
	1.0	0.0013	0.0002	0.0000	0.0000	0.0000	0.0007	0.0000	0.0000
All_VB_3SW	0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0008	0.0005	0.0000
	0.2	0.0000	0.0000	0.0000	0.0000	0.1043	0.0295	0.0110	0.0150
	0.4	0.0000	0.0000	0.0000	0.0000	2.8210	5.3817	0.7099	1.6883
	0.6	0.0000	0.0000	0.0000	0.0000	8.6694	2.7382	2.1512	0.1012
	0.8	0.0000	0.0000	0.0000	0.0000	0.1238	0.2734	0.0000	0.0000
	1.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0007	0.0026	0.0000
RCL_VB_3SW	0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0008	0.0005	0.0000
	0.2	0.0008	0.0000	0.0000	0.0000	0.0858	0.0295	0.0110	0.0150
	0.4	0.0000	0.0000	0.0000	0.0000	3.2376	5.2712	0.5842	1.7702
	0.6	0.0000	0.0000	0.0000	0.0000	7.8860	2.6756	2.1241	0.5020
	0.8	0.0000	0.0000	0.0000	0.0000	0.1319	0.2840	0.0000	0.0000
	1.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0011	0.0000	0.0000

Recent FEP Working Papers

Nº 285	Patricia Teixeira Lopes and Rui Couto Viana, "*The transition to IFRS: disclosures by Portuguese listed companies"*, August 2008
Nº 284	Argentino Pessoa, "*Educational Reform in Developing Countries: Private Involvement and Partnerships"*, July 2008
Nº 283	Pedro Rui Mazeda Gil and Óscar Afonso, "*Technological-Knowledge Dynamics in Lab-Equipment Models of Quality Ladders"*, July 2008
Nº 282	Filipe J. Sousa and Luís M. de Castro, "*How is the relationship significance brought about? A critical realist approach"*, July 2008
Nº 281	Paula Neto; António Brandão and António Cerqueira, "*The Macroeconomic Determinants of Cross Border Mergers and Acquisitions and Greenfield Investments"*, June 2008
Nº 280	Octávio Figueiredo, Paulo Guimarães and Douglas Woodward, "*Vertical Disintegration in Marshallian Industrial Districts"*, June 2008
Nº 279	Jorge M. S. Valente, "*Beam search heuristics for quadratic earliness and tardiness scheduling"*, June 2008
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

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