Pergamon

PII: S0305-0548(96)00052-4

Computers Ops Res. Vol. 24, No. 3, pp. 199–208, 1997

0305-0548/97 $17.00+0.00

USING TABU SEARCH TO SOLVE THE COMMON DUE DATE EARLY/TARDY MACHINE SCHEDULING PROBLEM

R. J. W. James†‡

Department of Management, University of Canterbury, Private Bag 4800, Christchurch, New Zealand

(Received September 1995; in revised form June 1996)

Scope and Purpose—This article examines the problem of scheduling jobs which need to be completed on one specified date. The objective is to minimize costs, where different costs are incurred if jobs are completed before the due date or after the due date. The problem is solved by using a search heuristic, tabu search. Various tabu search methods are tested. The most efficient methods exploit the known properties of optimal solutions to this problem so that only schedules that adhere to these properties and are feasible with respect to the earliest start time are ever considered in the search. This result provides new insights into the role of solution spaces and neighbourhood schemes when using search based solution techniques.

Abstract—This article uses tabu search to solve the restricted, common-due-date, early/tardy machine scheduling problem generalised earliness and tardiness penalties. Different forms of the tabu search are tested, including one based on a sequence of jobs solution space and another based on an early/tardy solution space. Results show that a search which uses an early/tardy solution space with a neighbourhood scheme which eliminates infeasible areas of the solution space is the most efficient and effective solution method.

1. INTRODUCTION

Early/tardy (E/T) Scheduling problems attempt to model just-in-time (JIT) type scheduling environments where producing orders early, as well as late, is discouraged. In a JIT scheduling environment customers may not accept early delivery of goods, even if they do they may not be willing to be billed for the goods until the original due date. In either case, the firm inccurs several costs when an order is produced early; for example, costs caused by the extra investment in finished goods inventory, costs involved with extra storage facilities and the cost of product spoilage.

The majority of the literature on the E/T scheduling problem has dealt with one particular class of problem, known as the common due date (CDD) problem, where all jobs are due to customers on the same date. There are many different forms of this problem proposed in the literature, each derived from differing assumptions about the due date and/or the earliness and tardiness penalties for each job.

The most general form of this problem, which we consider in this article, has no assumptions on the due date, penalties or processing times. For this problem we wish to minimise the total cost function which is expressed as follows:

$$\sum_{i=1}^{n}\left(\alpha_i|d-c_i|^{+}+\beta_i|c_i-d|^{+}\right)$$

where

α_i=The cost per unit of time incurred if job i is produced early — referred to as the earliness penalty.
β_i=The cost per unit of time incurred if job i is produced late — referred to as the tardiness penalty.
c_i=The time when job i is completed.
d=The common due date.
n=The number of jobs to be scheduled.
$|x|^{+}=x$ if $(x>0)$, 0 otherwise.

One assumption often made about the due date is that it is sufficiently large so that it does not constrain

† email: r.james@mang.canterbury.ac.nz
‡ Graduated from the University of Waikato with a Bachelor of Management Studies and a D.Phil. He is currently lecturing Productions and Operations Management and Operations Research at the University of Canterbury. His research interests include early/tardy scheduling problems, search heuristics and early/tardy penalty determination.

the scheduling of jobs, that is:

$$d \geq \sum_{i=1}^{n} p_i,$$

where p_i= the processing time of job i.This is known as the 'unrestricted' version of the CDD problem. The 'restricted' form of this problem places no such constraints on the due date value.

In their review of E/T scheduling, Baker and Scudder [1] note that the unrestricted common due date problem with job dependent penalties, that is the problem where there are no constraints on the value of either the earliness or tardiness penalty for any job, has been researched but that 'the restricted version of these problems has not been addressed'. Since their review some further work has appeared for the unrestricted problem with job dependent penalties[2], and other work has appeared for the restricted problem which does not allow job dependent penalties[3]. The restricted problem with job dependent penalties has still not been addressed due to the highly non-regular nature of the objective function which inhibits development of pseudo-polynomial time algorithms. The problem that is the focus of this paper is the generalised CDD problem which may or may not be restricted by its due date. We therefore need to consider in detail the properties of the restricted problem and regard the unrestricted problem as a special case of the restricted problem.

The popularity of the unrestricted form of the problem is due, in part, to the unique properties which assist in its solution [1]. The properties of the unrestricted form of the problem are:

(1) No idle time is inserted within the schedule.
(2) The schedule is 'V-shaped'; that is, early jobs are sequenced in non-increasing order of the ratio p_i/α_i and late jobs are sequenced in non-decreasing order of the ratio p_i/β_i, see Fig. 1.
(3) One job is completed on the due date.
(4) The b^{th} job in the sequence is completed on the due date, where b is the smallest integer satisfying the inequality:

$$\sum_{j=1}^{b} (\alpha_j+\beta_j) \geq \sum_{j=1}^{n} \beta_j.$$

The proof of properties 1 and 2 for the unrestricted case are general enough to also hold for the restricted case as well, however properties 3 and 4 will not hold in the restricted case. [3] proved that when $\sum_{j=1}^{b} p_j > d$, that is the problem is restricted, then the optimal schedule will start immediately, which we will refer to as time '0'.These properties effectively define the sequence jobs must adhere to for an optimal schedule. The problem therefore becomes one of determining which jobs will be scheduled early in the schedule and which ones will be scheduled late.

Section 2 discusses how Tabu search has been used to solve other scheduling problems. Section 3 develops a new neighbourhood scheme, while Section 4 outlines how the Tabu Search technique was implemented to solve the Restricted CDD Early/Tardy Scheduling problem. Section 5 details the computational experiments that were performed to test how well these searches performed, while Section 6 outlines some of the conclusions that can be drawn from the results of these experiments.

2. SCHEDULING USING TABU SEARCH

Tabu Search (TS) was developed by Glover [4–6] as a search technique for solving a wide variety of NP-hard problems. Glover [5] refers to tabu search as a 'meta-heuristic' since it is a technique which is

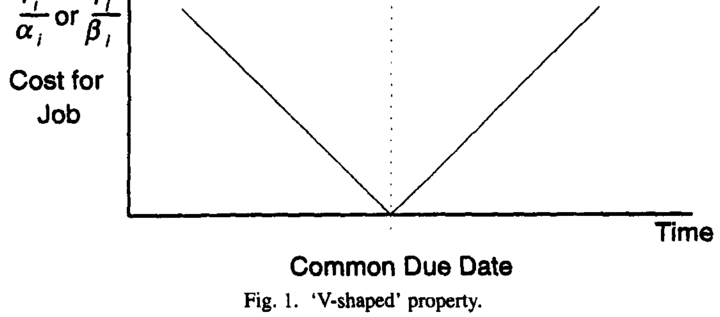

Fig. 1. 'V-shaped' property.

superimposed on a heuristic but is not, in itself, a heuristic. The heuristic used in conjunction with tabu search defines the solution space and the neighbourhood for the search. The neighbourhood defines all possible moves from the current point in the solution space. Tabu search chooses the next solution by evaluating either all or a set of neighbours to the current solution and moves to the neighbour having the best objective value. The move made by the heuristic then becomes 'tabu' and cannot be reversed until either a predefined number of iterations have passed or the resulting solution obtained by reversing this move meets some 'aspiration criteria' which is usually that the solution obtained by the move is better than the best solution found to date.

The tabu search techniques that have been applied to scheduling problems have commonly adopted the same neighbourhood schemes as travelling salesman problems [7]. Three common neighbourhood schemes for scheduling problems are:

(1) Adjacent Pairwise Interchange, where a job may be swapped with jobs directly to its left or right in the schedule.
(2) Swap (or All Pairwise Interchange), where any two jobs in the schedule can be swapped.
(3) Insert, where any job can be inserted in front of any other job in the schedule.

Comparative studies of these three neighbourhood schemes have shown that the insert neighbourhood scheme produces consistently better results than the swap neighbourhood scheme [8] and that the swap neighbourhood scheme produces consistently better results than the adjacent pairwise interchange neighbourhood scheme [9]. The best neighbourhood schemes, however, are a hybrid of the swap and insert neighbourhoods [7].

Reeves [10] incorporated a randomised candidate selection procedure in his implementation of tabu search which was used to solve the flowshop sequencing problem. This candidate selection procedure randomly orders the jobs and, from this random ordering, selects the first job and evaluates its neighbours. It continues with the next job until a specified number of jobs have been processed. The ratio of the number of jobs to process to the total number of jobs, is referred to as the 'window size'. As the window size is reduced, the search moves to new solutions faster than the standard tabu search, especially when large scheduling problems are being solved, however the quality of the move is unlikely to be as high. There is therefore a time/quality tradeoff in the problem which is dependent on the number of jobs in the problem being solved.

Hao *et al.* [11] used tabu search to solve a similar scheduling problem, the common due date assignment and sequencing problem. This problem does not have the due date specified as an input to the problem and therefore in terms of scheduling is an unrestricted CDD problem. This means that all the properties outlined in Section 1 can be applied. A further assumption of the problem addressed [11] is that the earliness and tardiness penalties are the same for each job which is not the case for the problem examined here.

3. SOLUTION SPACE DEVELOPMENT

Although tabu search has been used in a number of scheduling applications, (see [7] for a review) none have addressed the restricted common due date problem with job dependent earliness and tardiness penalties. The majority of the TS scheduling literature uses a solution space based on the physical sequence of jobs, which we will refer to as the sequence of jobs solution space, as this enables the use of neighbourhood schemes such as the swap and insert schemes described in Section 2. When this type of solution space is used with a problem such as the restricted CDD E/T scheduling problem, many of the solutions it produces are known to be suboptimal as they do not meet the V-shaped optimality condition outlined in Section 1.

An alternative approach is to avoid searching these sub-optimal solutions and only examine solutions that are V-shaped, as proposed [11]. We do this by using a binary solution space, which designates each job as being either early or tardy and, by using the algorithm outlined in Fig. 2, a schedule of jobs can be formed. To enumerate the sequence of jobs solution space, $n!$ job permutations would need to be examined. To enumerate the binary solution space, 2^n early/tardy job settings need to be examined. By using the binary solution space scheme the solution space size reduces from $n!$ to 2^n which is substantial as n becomes larger. Using the binary solution space ensures that all solutions which are being searched are likely to be of a higher quality as they are always adhere to the V-shaped property which means it is always optimal in terms of job sequence on either side of the due date. This solution space will be referred to as the E/T solution space.

1. Move all early jobs to the start of the sequence
2. Move all tardy jobs to the end of the sequence
3. Sequence all Early jobs in non-increasing order of p_i/α_i
4. Sequence all Tardy jobs in non-decreasing order of p_i/β_i
5. Set b = the number of jobs defined as early.
6. if $\sum_{j=1}^{b} p_j > d$ then this E/T job specification is infeasible as the processing time required to complete all the early jobs is greater that that available due date - therefore return an error.
7. Set the completion time $c_b = d$. Calculate all the remaining completion times in the schedule from this job using the relationship that $c_i = c_{i+1} - p_{i+1}$.

Fig. 2. E/T job specification to schedule conversion algorithm.

Using the E/T solution space, some of the solutions obtained will be infeasible if the sum of the processing times of all the jobs which have been designated as being early is greater than the due date. Despite this, the solution space is still guaranteed to be fully connected, that is if the search starts at one point in the solution space it is guaranteed that there is a route from that point to every other point in the solution space, including the optimal solution. This can be easily proven by considering that each valid schedule is always connected to another valid schedule which has one less early job in it. By repeating this process, all valid schedules degenerate into the schedule where all jobs are scheduled late. Since all schedules degenerate into this one schedule, any point in the solution space, including the optimal solution, must be reachable from any other point.

4. TABU SEARCH IMPLEMENTATION

To compare the effect of different solution spaces on the effectiveness of the tabu search technique, two different types of tabu searches were implemented — one based on a sequence of jobs solution space, the other based on an early/tardy solution space.

4.1. Sequence of jobs based tabu search techniques

The base TS algorithm uses the sequence of jobs solution space with a combination of both insert and swap moves being allowed. A fixed length tabu list is also used with the aspiration criteria that a tabu move can be accepted if it has an objective function value with a cost lower than the best cost found so far. The tabu conditions used specify that the job moved in the insert neighbourhood scheme, and one of the jobs moved in the swap neighbourhood scheme are deemed 'tabu' and placed on the tabu list. This tabu job cannot move from its current position in the schedule for a fixed number of iterations unless the aspiration criteria is met. The stopping criteria terminates the search after a given amount of time. This enables a fair comparison of solution spaces which have different iteration times.

Candidate selection procedures are necessary in large scheduling problems to allow the search to progress at a reasonable rate. Two different candidate selection procedures were used. The first uses the random candidate selection procedure proposed [10], and is referred to as the Reeves-CL. The second scheme again uses the random candidate selection procedure [10] but narrows the candidates down further by restricting the distance that each job can move. This candidate selection procedure was used by Laguna *et al.* [12] and [13] in their implementations of TS for scheduling. The TS with both these candidate selection procedures implemented is referred to as the Reeves-Window-CL.

For example, suppose we have the sequence of jobs 1-2-3-4-5-6-7-8-9-10. If we have a candidate selection window of 3 jobs, then by using the candidate selection procedure proposed by Reeves we would generate a random selection of possible candidates, of say 8, 5, 10, 2, 4, 7, 3, 9, 6, 1. In our first iteration we would consider all swap and insert moves involving jobs 8, 5 and 10. In our second iteration we would consider moves involving 2, 4 and 7 and in the third iteration jobs 3, 9 and 6 would be considered. The fourth iteration requires a further list to be regenerated and would involve considering job 1 and the first two jobs in the new list.

When using the window of potential jobs candidate selection procedure, of 4 jobs, the insert and swap moves can be with jobs no more than 2 jobs away. Therefore when considering job 8, we can consider swap and insert moves with jobs 6, 7, 9 and 10.

The Reeves-CL uses a starting point which places the jobs in non-increasing order of p_i/α_i then schedules this sequence. The Reeves-Window-CL uses this same sequence but then 'optimises' it according to the relevant common due date properties.

4.2. *Early/tardy based tabu search*

The E/T Tabu Search is implemented in a similar manner to the sequence of jobs TS but instead of using a sequence of jobs solution space, it uses a solution space based on whether a job is to be scheduled early or late — referred to as the E/T job specification. The neighbourhood scheme used takes each job in turn, changes its E/T job status from being early to late, or vice versa and then evaluates its new E/T job specification by using the algorithm in Fig. 2.

Since there may be invalid areas in the E/T solution space it would be possible for the search to become trapped by infeasible areas of the solution space and tabu restrictions. This may occur when a job with a large processing time dominates the space before the due date making it impossible for the remaining late jobs to become early. To overcome this problem a special two level tabu list was used which contains not only a list of moves that are tabu but also a record of the number of times the search has violated that tabu status. The search will, if there are no non-tabu moves that can be made, use the tabu job with the least number of tabu violations. This enables the search to back-track to an area in the solution space that has not been searched and proceed from there. Fig. 3 illustrates this situation and how the search handles it. Once the search has entered a place within the solution space where it can not make any further moves because of invalid solution spaces (move 4), it back tracks on tabu moves (moves 5 and 6) to allow it to get back to an area where it is not trapped (move 7).

The search uses a fixed sized tabu list and the same aspiration criteria as the standard TS, with a tabu job being accepted if the schedule produced has a cost lower than the best schedule so far.

The initial schedule sequences jobs in non-increasing order of ratio p_i/α_i and then calculates the timing of the sequence using properties 3 and 4 outlined in Section 1. All jobs scheduled before the common due date are given an early job specification, while all jobs scheduled after the common due date are given a tardy job specification. From this E/T job specification, the sequence is recalculated using the algorithm in Fig. 2.

Again the random candidate selection procedure, proposed [10] is also implemented on this search to provide an increase in speed when dealing with larger problem sizes. This TS is referred to as the E/T Reeves-CL.

4.3. *Alternative E/T tabu search neighbourhood schemes*

Initial experimentation using the two E/T based TS techniques indicated that they were performing poorly on problems which had a high probability (greater than 50%) that jobs were going to be tardy. In

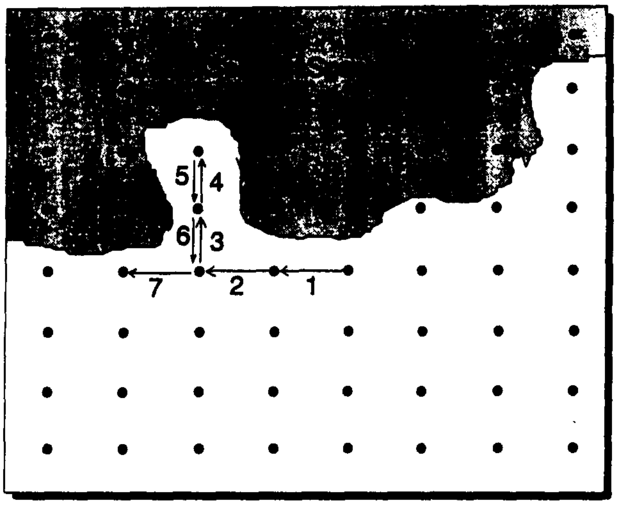

Fig. 3. Searching with invalid solution space.

these problems the due date is very constraining, that is the majority of jobs have to be scheduled late because the due date is too early. The progress of the search was very slow and tended to level off after a certain amount of time. Further investigations revealed that for these problems, the neighbourhood scheme that flips one job from being early to being late or vice versa, was not adequate. Effectively what happens is that once the space up to the due date becomes filled the only valid moves remaining require the early jobs to become late — freeing up some of the early job space enabling a late job to be moved to fill the space. This process becomes extremely cumbersome for the search. To overcome this problem, two new tabu searches were developed using different neighbourhood schemes.

The first technique is implemented in the same manner as the E/T Reeves-CL except that it uses an extended neighbourhood scheme. Instead of changing one job from early to late or vice versa, it also allows moves which change one job from early to late and another job to from late to early. This TS is referred to as the E/T Swap. All other aspects of this search are identical to the Reeves-CL E/T.

The second technique developed was called the E/T Date Swap. This search incorporates into the scheduling algorithm rules which will enable it to overcome the problem of the total processing time of early jobs being greater than the time before the due date. This search was inspired by the method inherently used by tabu searches based on the sequence of jobs solution spaces, which was found to be quite efficient for problems requiring a high number of tardy jobs. This method creates extra room by scheduling the early jobs closest to the due date late. Effectively what was incorporated into the E/T Date Swap was a diversification strategy which eliminates all invalid areas in the solution space, so that when the search encounters an invalid area it will automatically bring it into a new area of the solution space. The algorithm to achieve this is outlined in Fig. 4. All other aspects of this search are identical to the E/T Reeves-CL.

5. COMPUTATIONAL EXPERIMENT

5.1. Data generation

No standard E/T scheduling problem data has been reported in the literature, therefore problem data was generated using the method described by Baker and Martin [14]. This data generation method requires two parameters, the tardiness factor and the due date range. Since the due date range is irrelevant for common due date problems, as it requires only one due date, the entire problem data set can be defined by the tardiness factor:

The Tardiness Factor is the proportion of jobs which are expected to be tardy in a given sequence. Baker and Martin [14] define the tardiness factor as:

$$t=1-\bar{d}/n\bar{p}$$

where

$\bar{d}$=average due date (the common due date in this case), and
$\bar{p}$=average processing time of the jobs being scheduled.

---

1. **Move all early jobs to the start of the sequence**
2. **Move all tardy jobs to the end of the sequence**
3. **Sequence all Early jobs in non-increasing order of p_i/α_i**
4. **Sequence all Tardy jobs in non-decreasing order of p_i/β_i**
5. **For the resulting sequence, determine job b where b is the lowest integer satisfying $\sum_{j=1}^{b}(\alpha_j+\beta_j)\geq\sum_{j=1}^{n}\beta_j$**
6. **If $\sum_{j=1}^{b}p_j>d$ then set the completion time of job b at $\sum_{j=1}^{b}p_j$ otherwise set the completion time at d. Calculate all the remaining completion times in the schedule from this job using the relationship that $c_i=c_{i+1}-p_{i+1}$.**

Fig. 4. E/T job specification to schedule date swap conversion algorithm.

Table 1. Parameters for each class of problem

Class name	Tardiness factor	Problem size (Jobs)	Number of problems	Early penalty range	Tardy penalty range
LoT	0.2	250	30	[1,5]	[6,10]
HiT	0.6	250	30	[1,5]	[6,10]

Once the processing times for the jobs have been generated and a tardiness factor has been selected, the common due date to be used can easily be determined.

As suggested by Baker and Martin, the tardiness factors used in our tests were 0.2 and 0.6. This broke our problems into two distinct classes, ones with a low tardiness factor (LoT) and ones with a high tardiness factor (HiT). Two additional parameters were also included; the range of the earliness penalty, α, and the range of the lateness penalty, β. The range of values for this data was drawn from a discrete uniform distribution with $\alpha=[1,5]$ $\beta=[6,10]$. The earliness penalties are less than the tardiness penalties as this is the most likely case for businesses which are highly customer focused. Processing times were drawn from a normal distribution with a mean of 100 and a standard deviation of 25. Thirty problems from each class were generated.

Table 1 outlines the parameters used in each of the two problem classes used.

To determine whether an E/T solution space is beneficial to the solution of scheduling problems, each of the tabu search methods was used to solve each of the 60 common due date problems. Each technique was given 7500 CPU seconds on a VAX 6000 mainframe to solve each problem.

5.2. Experiments

The parameters for the TS were the best parameters found by [15] which used these search techniques for solving the distinct due date problem. These parameters are outlined in Table 2.

Figure 5 shows the average deviation from the known minimums for the problems generated above for each of the different TS techniques at the average time taken to find this minimum. The lines graphed represent the average results from the 60 different problems in terms of time and deviation. ANOVA was

Table 2. Tabu search parameters

	Tabu list size	Reeves-CL window size	Distance window size
Reeves-CL	15	0.01	N/A
E/T TS	15	N/A	N/A
E/T Reeves-CL	1	0.10	N/A
E/T Swap	7	0.10	N/A
E/T Date Swap	7	0.10	N/A
Reeves-Window-CL	15	0.20	0.04

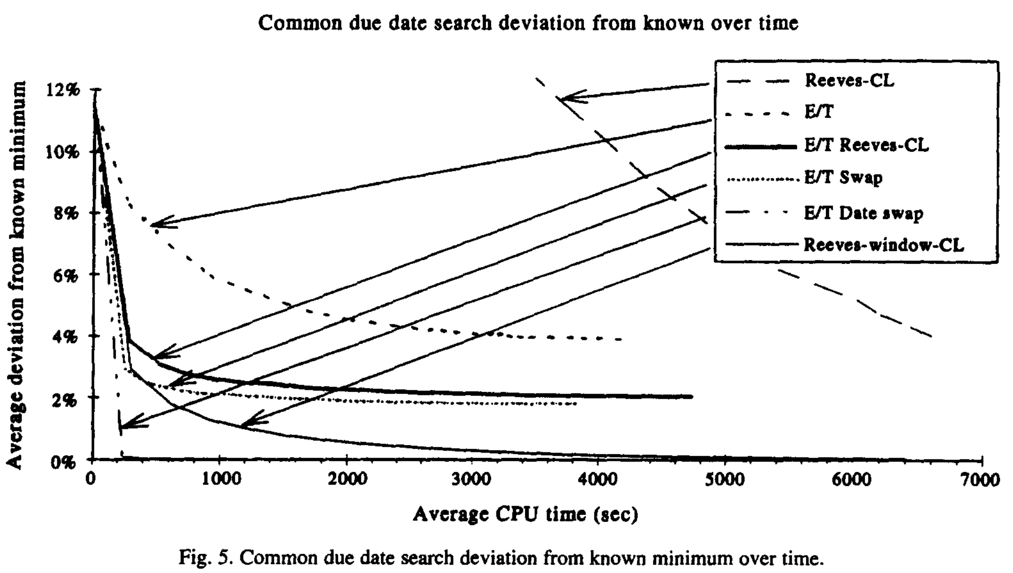

Fig. 5. Common due date search deviation from known minimum over time.

used to determine whether the performance of each method was significantly different. The E/T Date Swap method was significantly better than all of the other techniques at every point in time except the starting point, which was the same for all techniques except for the Reeves-CL. The E/T Reeves-CL technique and the Reeves-Window-CL were not significantly different for the first 300 CPU seconds, however after this the Reeves-Window CL technique proved to be superior.

Figure 5 emphasises the importance of using high quality starting points for these types of search techniques. The Reeves-CL takes over 3000 CPU seconds to obtain solutions of a similar quality to the starting point obtained from the E/T solution space. This in turn means lower quality solutions when time restraints are in place.

The E/T Reeves-CL is clearly more efficient than the E/T TS, indicating the benefit of implementing a restricted candidate selection criteria. The E/T Reeves-CL, however, was not superior to the Reeves-Window-CL which used the sequence of jobs solution space.

An interesting feature of the E/T Reeves-CL is the way in which the search quickly finds solutions which are less than 4% deviation from the known minimum, however after this initial gain, very little progress is made. The E/T Swap also follows a similar pattern. While the swap extension initially finds better solutions, after around 400 CPU seconds the Reeves-Window-CL starts to provide better quality solutions. This is caused by the swap strategy intensifying the search in a particular area of the solution space, which is not always beneficial to the search, especially when the search actually needs to be diversified into new areas of the solution space. The E/T Reeves-CL search is therefore more likely to explore different regions of the solution space, hence slightly better long term progress is made.

The best search technique was clearly the E/T Date Swap which, like the E/T Reeves-CL, also improved significantly very quickly, however this time the search found solutions close to the known minimums for each problem within an average time of 400 CPU seconds. This type of search has the same advantages as the E/T Swap, in that its performance is not affected by high tardiness problems, which have little space for scheduling early jobs. The added bonus of this neighbourhood scheme is that there are no infeasible areas in the solution space — hence the search is able to move freely to new areas without hindrance.

The Reeves-CL is significantly worse than all the other TS methods mainly because of its starting point. When the best sequence of jobs based TS method, the Reeves-Window-CL, and the best E/T job specification based TS, E/T CDD Date Swap, are compared it can be seen that these two methods are statistically different up until approximately 1000 CPU seconds at the 5% level of significance. This leads us to conclude that the use of an E/T solution space with TS to solve the restricted CDD problem can increase the speed of a search considerably with an appropriate choice of neighbourhood scheme.

Another fact reflected in the ANOVA was a strong difference in solution quality between the classes of problems being solved by the Reeves-Window-CL. Figure 6 shows the effect of solving problems with different tardiness factors on the solutions obtained by the both the Reeves-Window-CL and the E/T

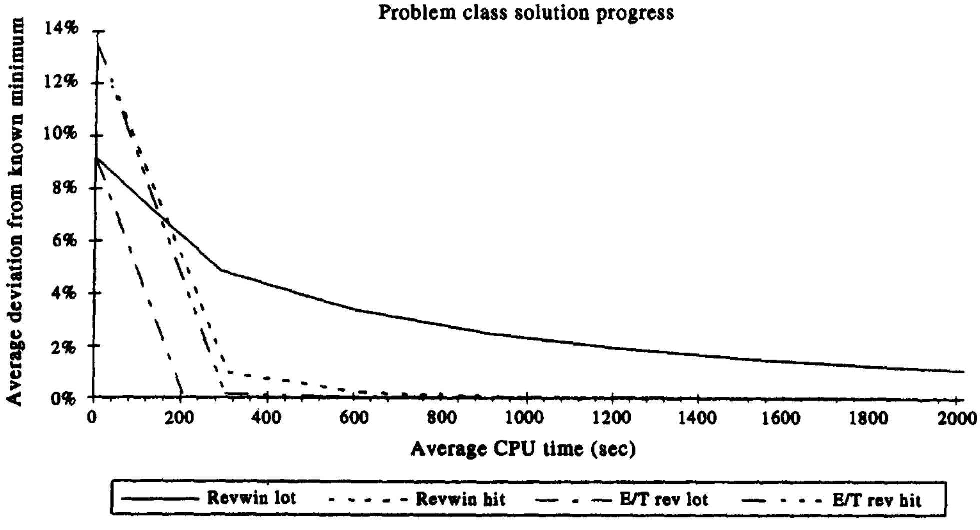

Fig. 6. Problem class solution progress.

CDD Date Swap method.

Figure 6 shows that the Reeves-Window-CL is slower at finding high quality solutions for problem classes with a low tardiness factor than problems classes with a high tardiness factor. The reason for a low tardiness factor slowing down the search would be caused by the less extreme costs in the neighbouring solutions of these problems, hence the solution space tends to be flatter and therefore more difficult to search.

A one way ANOVA was carried out to determine whether there was a significant difference between the results obtained from the problems with a high tardiness factor and those with a low tardiness factor. Each of the searches produced results which differ significantly with the tardiness factor of the problem. From Fig. 6 The E/T Date Swap appears to solve the problem classes with a lower tardiness factor slightly faster than the those with a higher tardiness factor. The reason for this is that when the tardiness factor is low, the heuristic is less likely to reschedule early jobs when testing neighbours which change a late job to an early job, which is a more common occurrence when the tardiness factor is high.

6. CONCLUSIONS

From these experiments it can be concluded that the use of an E/T solution space can increase the performance of search techniques, such as TS, when compared to using a sequence of jobs solution space. This is especially true for problems that have a low tardiness factor. The main reason for this gain is that when TS uses the E/T solution space it is searching only high quality results which do not have many of the less efficient solutions included. This means that the search can move more quickly to good quality solutions. This proves the worth of incorporating problem specific 'knowledge' about the solution into the search rather than just using general formulations. The E/T solution space provides a convenient method of representing this knowledge for the Restricted CDD E/T scheduling problem.

Overall the E/T Date Swap was found to give the best performance on average of all the search techniques. The standard E/T TS suffered when the tardiness factor was high as many of the moves which involved scheduling late jobs early became infeasible, making it difficult for the search to move to new areas of the solution space. This experiment also highlights an important fact when considering the design of new solution spaces and neighbourhood schemes. Solution spaces which have invalid areas, may hinder the search greatly. Implementing an automated diversification strategy, in the form of the E/T Date Swap algorithm, provides a way around invalid solutions and, as demonstrated, significant performance gains can be achieved.

The experiments here show that substantial gains can be made by a search technique with an E/T solution space, at least initially. A standard search can utilise these initial gains by using the result as a starting point for its search, in effect creating a hybrid technique. The E/T solution space would be used to find the most suitable area to search while the standard TS would be used to 'fine tune' the solution.

Acknowledgements—The author would like to thank Dr. John Buchanan for his assistance in the preparation of this paper.

REFERENCES

1. Baker, K. R. and Scudder, G. D., Sequencing with earliness and tardiness penalties: a review. *Ops Res.*, 1990, **38**, 22–36.
2. Hall, N. G. and Posner, M. E., Earliness-tardiness scheduling problems, I: weighted deviation of completion times about a common due date. *Ops Res.*, 1991, **39**, 836–846.
3. Hall, N. G., Kubiak, W. and Sethi, S. P., Earliness-tardiness scheduling problems, II: deviation of completion times about a restrictive common due date. *Ops Res.*, 1991, **39**, 847–856.
4. Glover, F., Future paths for integer programming and links to artificial intelligence. *Comp. Ops Res*, 1986, **15**, 533–549.
5. Glover, F., Tabu search - Part I. *ORSA J. Comp.*, 1989, **1**, 190–206.
6. Glover, F. and Laguna, M., Tabu Search. In *Modern Heuristic Techniques for Combinatorial Problems*, ed. C. R. Reeves. Blackwell Scientific Publications, 1993, pp. 70–150.
7. Barnes, J. W. and Laguna, M., A tabu search experience in production scheduling. *Annals Ops Res.*, 1993, **41**, 141–156.
8. Laguna, M. and Glover, F., Integrated target analysis and tabu search for improved scheduling systems. *Expert Systems with Applications*, 1993, **6**, 287–297.
9. Tsubakitani, S. and Evans, J. R., Applying tabu search to the mean tardiness sequencing problem. Working Paper, University of Cincinnati, April, 1992.
10. Reeves, C. R., Improving the efficiency of tabu search for machine sequencing problems. *J. Op. Res. Soc.*, 1993, **44**, 375–382.
11. Hao, Q., Yang, Z., Wang, D. and Li, Z., Common due-date determination and sequencing using tabu search. *Comp. Ops Res.*, 1996, **23**, 409–417.
12. Laguna, M., Barnes, J. W. and Glover, F., Tabu search methods for a single machine scheduling problem. *J. Intelligent Manufact.*, 1991, **2**, 63–74.
13. Laguna, M. and González Velarde, J. L., A search heuristic for just-in-time scheduling in parallel machines. *J. Intelligent Manufact.*, 1991, **2**, 253–260.
14. Baker, K. R. and Martin, J. B., An experimental comparison of solution algorithms for the single-machine tardiness problem.

Naval Res. Logist. Quart., 1974, **21**, 187–199.
15. James, R. J. W. and Buchanan, J. T., A neighbourhood scheme with a compressed solution space for the early/tardy scheduling problem. *Eur. J. Op. Res.*(in press).