
Single machine scheduling for minimising earliness and tardiness penalties by scatter search approach

M. Saravanan*

PVP College of Engineering & Technology for Women,
K. Sinkarakottai, Dindigul-Batlagundu Highways,
Dindigul, Tamil Nadu, 624 702, India
E-mail: sarandgl2k@yahoo.co.in
*Corresponding author

A. Noorul Haq

Department of Production Engineering,
National Institute of Technology,
Tiruchirappalli, Tamil Nadu, India
E-mail: anhaq@nitt.edu

Abstract: This paper proposes a new evolutionary technique named scatter search for scheduling a number of jobs on a single machine against a restrictive common due date. Individual earliness and tardiness penalties for the jobs are under consideration and the objective is to find an optimal schedule, which jointly minimises the sum of the earliness and tardiness costs. In this paper, the results are obtained for numerous benchmark problems generated by Biskup and Feldmann (2001) for common due date (CDD) problems. The best results of each of the benchmark problems of the three meta-heuristic techniques used by Feldmann and Biskup (2003) are selected and they are compared with the results of SS. This methodology provides substantially better results than the benchmark problems. At the same time, this method holds almost the same result as the best results (best-FB) of three meta-heuristic approaches of Feldmann and Biskup (2003).

Keywords: scheduling; metaheuristic; scatter search; SS; tardiness; earliness; benchmark; common due date; CCD.

Reference to this paper should be made as follows: Saravanan, M. and Noorul Haq, A. (2011) 'Single machine scheduling for minimising earliness and tardiness penalties by scatter search approach', *Int. J. Electronic Transport*, Vol. 1, No. 1, pp.5–25.

Biographical notes: M. Saravanan is the Principal of the PVP College of Engineering & Technology for Women, K. Sinkarakottai, Dindigul, Tamil Nadu, India. Previously, he was Professor and Head at the Department of Mechanical Engineering at R.V.S. College of Engineering and Technology, Dindigul, India. He completed his PhD in the field of scheduling optimisation for various manufacturing systems at Anna University, India. His research interests are in the areas of scheduling for manufacturing systems, robotics, production planning, optimisation techniques, etc. He received his BE in Mechanical Engineering from Madurai Kamaraj University and ME in Production Engineering from Government College of Technology (Bharathiar University), Coimbatore, India. He has published 19 technical papers in

refereed international journals and more than 62 papers in national and international conferences. He has bagged three university awards and a state award for his tireless services in national service scheme for more than 12 years.

A. Noorul Haq is a Professor of Production Engineering Department at the National Institute of Technology, Tiruchirappalli, India. He received his BE in Mechanical Engineering from Annamalai University in 1975, ME in Mechanical Engineering from Madras University in 1981 and PhD in Manufacturing Management from IIT, New Delhi, India in 1991. His research interests are in the areas of aggregate production planning, logistics and supply chain management, operations management, vehicle routing, scheduling, inventory control and non-traditional optimisation techniques. He has published about 90 technical papers in refereed journals and more than 100 papers in conferences.

1 Introduction

The operations scheduling problems have been studied for over five decades. One of the most thoroughly studied scheduling problems is the single machine scheduling problem (SMSP). In the SMSP, a set $N = \{1, \dots, n\}$ of n independent jobs has to be processed on one machine. SMSPs make the basis for theories of scheduling. There are several reasons for focusing on single machine problems. However, since SMSPs bear complex computations, it seems that the analysis of such problems is the most cost-effective for a better understanding of them and their intrinsic structure. In many cases, it is single machine problems that lead into bottleneck in a production environment. Among single machine problems, those related to earliness and tardiness penalties are more important. In real world, since a customer expects to receive the product on a specific date, scheduling based on the due date is an important task in the production plants. Methods such as lean management, simultaneous engineering and just in time (JIT) production consider such needs. One of the most important production systems to satisfy customer requirements is JIT system. This system shows a situation where a given quantity of product must be produced and delivered to the customer in a specified time. In JIT production, the due date could be considered as the common due date (e.g., a series of jobs requiring simultaneous mounting at a higher stage of production). In scheduling problems, considering the common due date (CCD), some jobs may be completed earlier and some others later than the due date, making penalties in both cases. Earliness leads into inventory and maintenance penalties while tardiness leads into dissatisfaction of a customer and losing goodwill and reputation. In the field of CCD scheduling, two kinds of due dates, namely, restrictive and unrestrictive ones, are distinguished: a CCD is called unrestrictive, if its optimal value has to be determined, or if it is given and has no influence on the optimal sequence. Note that a given due date which is greater than or equal to the sum of processing times of all jobs is always unrestrictive. If, on the other hand, a CCD is given and may influence the optimal sequence of the jobs, it is called restrictive; thus a search for an optimal sequence has to be carried out with respect to the due date. Scheduling on a single machine is a special case of the multiple-machine environment. There are a number of reasons for focusing on single-machine problems: multiple-machine scheduling problems are, as a rule, computationally intractable and a

better understanding of them and their inherent structure may be gained by analysing single-machine problems. Furthermore, in many real life situations it is one machine that causes a bottleneck of the whole production environment, so that production planning should be orientated towards this single machine. Therefore, the analysis of restricted-type problems seems necessary here. This type of problems has been studied in the literature since long ago.

When scheduling on a single machine about a CCD, at most one job can be completed exactly on the due date. Other jobs have to be completed either earlier or later than the due date. To meet the just-in-time requirement, jobs completed before the CCD incur earliness penalties, while jobs completed after the CCD incur tardiness penalties. The earliness penalties include holding costs for finished goods, deterioration of perishable goods and opportunity cost. The tardiness penalties are the back logging cost which includes performance penalties, loss of sales and loss of goodwill. The single-machine scheduling problem with general earliness-tardiness penalties about a CCD is defined as follows. There are n jobs each of which requires exactly one operation are scheduled on a single machine about the CCD d . For each job i , the processing time p_i , the penalty per unit time of earliness α_i , and the penalty per unit time of tardiness β_i are given.

2 Literature review

Scheduling against due dates has been receiving considerable attention in the literature. Conway (1965) was the first person who formally raised the question of the CCD as a part of scheduling and stated that this is the due date that could have different consequences. Sidney (1977) and Kanet (1981) were among other leading investigators on the due date problems. Sidney (1977) introduced an effective algorithm to represent the optimal date and studied distinct due dates. One of the pioneers studying CCD problems has been Kanet (1981), who considered the problem of minimising the sum of deviations from a CCD and presented a polynomially bounded matching algorithm which solves the problem in $O(n \log n)$ time. This contribution has been extended in many directions; see, for example, Biskup and Cheng (1999), Hall and Posner (1997), Hoogeveen and van de Velde (1991) and Panwalkar et al. (1982). An excellent review is given by Baker and Scudder (1990). Hall (1986) studied the investigations of Kanet (1981) on several parallel machines and analysed optimal conditions and optimal solutions available for such problems. A comprehensive review of the due date was made by Baker and Scudder (1990) and Gordon et al. (2002). These authors stated that the limited due date problems were not studied before them. Cheng and Gupta (1989) carried out a study to solve scheduling problems for parallel machines and indicated a review for decision-making on the due-date. The earliness-tardiness models, considering a given CCD, are classified as different types of penalty. Some of these models, like Emmons (1987), describe common penalties, some others, like Bagchi et al. (1987) and Panwalkar et al. (1992), have treated the difference between earliness-tardiness penalties (Emmons, 1987; Bagchi et al., 1987; Panwalkar et al., 1982). Another group noted the remaining jobs, including Baker and Scudder (1989) and Cheng (1987). Cheng (1990) studied different types of scheduling problems with CCD. The author considered the objective function only as a combination of jobs with earliness-tardiness penalties, where

earliness of jobs was penalised with the same rate. For more studies, the detailed study is reported in Weng and Ventura (1994). Webster and Gupta (1998), considering an arbitrary rate for earliness-tardiness penalties, studied the due date problems for this group of problems using the genetic algorithm (GA). Concerning limited due date problems, Hall and Posner (1991) showed that they are of NP-hard type. In the same direction, Lee and Kim (1995) raised the GA, while James (1997) applied the tabu search (TS) approach. Hall et al. (1991) showed this group of problems considering the objective function as minimising the sum of unweighted earliness and tardiness and considered the optimum conditions and the NP-hard argument for the problem. The method was raised by Hino et al. (2005) using the two methods of TS and GA to solve problems of CCD were appropriate in facilitating solution of limited due date problems. Also, the integer programming methods raised by Biskup and Feldmann (2001) for solving problems with small size of jobs were useful. The same authors studied SMSPs with limited CCD in 2003 and showed that the threshold accepting method is an appropriate method for this purpose (1, 2). Seyed and Mohammad (2008) analysed several meta-heuristic algorithms to solve restricted SMSPs and then propounded a heuristic method called tabu search. Using statistical analyses, they showed that the proposed algorithm is as efficient as the other methods of simulated annealing and threshold accepting.

This paper focuses solely on single-machine problems, assuming that a restrictive CCD exists and for each of the jobs, individual earliness and tardiness completion time penalties are given. The goal is to find a schedule, which minimises the sum of earliness and tardiness costs. This problem is NP-hard (Hall et al., 1991; Hoogeveen and van de Velde, 1991). Although the exact classification is an open question, no pseudo polynomial algorithm is known and it is conjectured that the problem is NP-hard in the strong sense (Lee et al., 1991). Consequently, it has been tackled by meta-heuristic approaches, namely, tabu search (James, 1997) and parallel genetic algorithms (Lee and Kim, 1995). As both approaches are marked by an underlying weakness, which was discussed in connection with the properties of the problems, Feldmann and Biskup (2003) developed a new and appropriate problem representation, which overcomes this weakness. In that paper, three meta-heuristics, evolutionary search (ES), simulated annealing (SA) and threshold accepting (TA) using a new problem representation was presented. In addition, a new variant of TA, namely 'TA with a back step', is introduced. The approaches are implemented and tested extensively on benchmark problems given by Biskup and Feldmann (2001) and James (1997). In this paper, a new evolutionary technique called SS with the same problem representation used by Feldmann and Biskup (2003) for optimising the problems is introduced and the results are compared with the best solution obtained from the above three methods. Noorul Haq and Saravanan (2007) used SS method for minimising the earliness and tardiness penalties in single machine scheduling for common due window (CDW) problems. The application of SS algorithm for other manufacturing system, performs better in scheduling problems such as flow shop [Noorul Haq et al., 2007; Saravanan and Noorul Haq, 2008], FMS problems (Saravanan and Noorul Haq, 2008), Single machine scheduling (Noorul Haq and Saravanan, 2007), cell scheduling (Saravanan and Noorul Haq, 2008), vehicle routing problem (Saravanan and Sundararaman, 2010) and job shop scheduling (Saravanan and Noorul Haq, 2010). Ying (2008) has implemented recovering beam search (RBS) algorithm to minimise earliness and tardiness penalties for CCD. Wang (2006) has presented an $O(n \log n)$ algorithm to obtain the optimal solution for single machine

scheduling with CCD. Since SS method gives almost the same result as the best solution obtained from the above three methods, our problems are restricted to the tune of 100 jobs.

The remaining part of the paper is structured as follows. In Section 3, the notation needed is introduced and the problem under study is formulated. Furthermore, some properties of this problem, which are used for the development of the problem representation in Section 4, are stated. The basic elements of SS based on the well-known ‘five methods’ are presented in Section 5. Computational results are summarised in Section 6 and the paper concludes with final remarks.

3 Problem formulation

Nomenclature

d common due date

n number of jobs

p_i processing time of job i

a_i earliness penalty of job i per time unit

b_i tardiness penalty of job i per time unit

C_i completion time of job i

E_i earliness of job i

T_i tardiness of job i

B set of non-tardy jobs, $B = \{i / C_i \leq d\}$

A set of tardy jobs scheduled to begin in or after d , $A = \{i / C_i - p_i \geq d\}$.

All jobs are assumed to be available at time zero. They have to be processed on a single machine and each of the jobs needs exactly one operation. The processing times p_i is deterministic and known in advance, preemption of jobs is prohibited and a restrictive CCD d is given. A job is early, if its completion time is smaller than the CCD. On the other hand, a job is tardy, if its processing ends after d . As it is not known in advance whether a job will be completed before or after the due date, earliness and tardiness are calculated as $E_i = \max \{d - C_i, 0\}$ and $T_i = \max \{C_i - d, 0\}$ for all jobs $i = 1, \dots, n$. The objective is to find a feasible schedule S which minimises the sum of earliness and tardiness penalties, is given as

$$f(S) = \sum_{i=1}^n \alpha_i E_i + \sum_{i=1}^n \beta_i T_i$$

We refer to the term ‘sequence’ as the order in which the jobs are processed. A ‘schedule’ on the other hand contains all the information which is needed to manufacture the jobs; these are the sequence, the starting time of the first job and the CCD. In the

following three well-known properties of the problem under study are presented. They are essential for the new problem representation and for the design of the heuristic approaches.

Property 1. In an optimal schedule there are no idle times between the processing of consecutive jobs; a general proof is given by Cheng and Kahlbacher (1991).

Property 2. An optimal schedule has the so-called V-shaped property. This means that the jobs $i \in B$ are ordered according to non-increasing ratios p_i / α_i and the jobs $i \in A$ are in non-decreasing ratios p_i / β_i ; the proof can be done by interchange argument, see for example, Baker and Scudder (1989).

Note that despite Property 2 an optimal schedule might contain a so-called straddling job not belonging to the sets A and B; in which processing is started before and finishes after the due date. If in an optimal schedule a straddling job j exists, no definite statement of its ratios p_j / α_j and p_j / β_j can be made, see Biskup and Feldmann (2001) and Hooegeveen and van de Velde (1991).

Property 3. An optimal schedule exists in which either the processing of the first job starts at time zero or one job is completed at the due date; the proof is similar to that of Hooegeveen and van de Velde (1991) for the restrictive CCD problem with weighted penalties. Szwarc (1989) has been the first to demonstrate that for the restricted CCD problem (with $\alpha_i = \beta_i = 1$) instances exist in which none of the optimal schedules start at time zero.

Note that Property 3 implies the possibility that in an optimal schedule the first job starts at time zero and the completion time of one job coincides with the due date. Nevertheless, considering Property 3 the search for an optimal schedule should not be restricted to sequences starting at time zero. The weakness of the two above-mentioned approaches of James (1997) and Lee and Kim (1995) is that both neglect Property 3 and limit their search space to schedules in which the first job starts at time zero although this might exclude all optimal schedules a priori.

4 Problem representation

The choice of an appropriate problem representation is among the most important tasks for the application of meta-heuristics. The problem representation can be seen as a mapping, which transfers a feasible solution of the problem into a specific, somehow coded, represented solution. As a matter of principle the problem representation should fulfill the following requirements, see Feldmann (1999) and Osman and Laporte (1996):

- it has to be complete, i.e., all possible solutions must be representable
- none of the represented solutions may be infeasible
- the mapping should only contain redundancies, if this is advantageous for the search process
- a given solution should be represented unequivocally

- with the application of a specific search operator each solution of the entire solution space must be reachable independently of the starting point
- the problem representation should in addition support the application of appropriate operators.

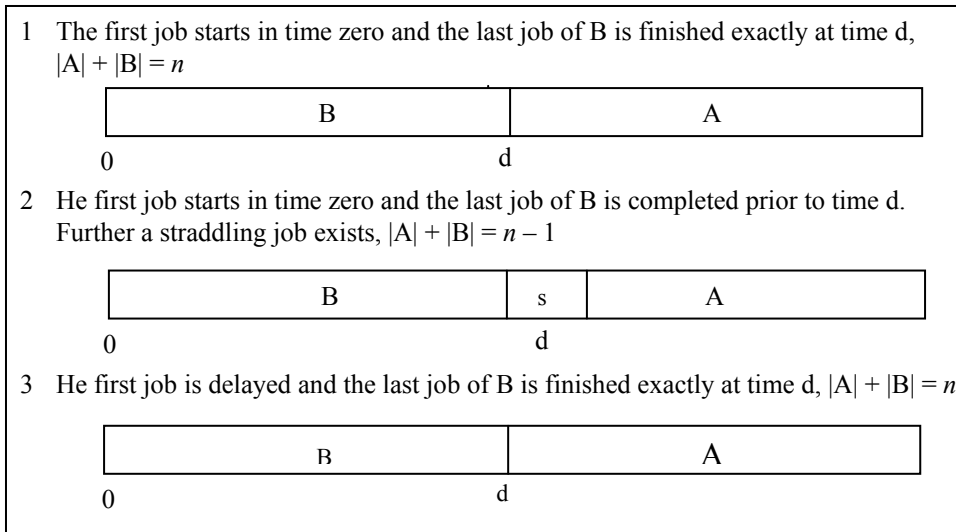
A representation for the CCD problem has to take into consideration its structural properties, i.e., a straddling job violating the V-shaped property might exist and the starting point of the first operation is not known in advance. In any event it is possible to calculate the maximum number of non-tardy jobs $|B|_{\max}$ for which the inequalities

$$\sum_{i=1}^{|B|_{\max}} P_i \leq d \quad \text{and} \quad \sum_{i=1}^{|B|_{\max}+1} P_i > d$$

hold, assuming that the jobs are indexed according to the shortest processing time (SPT) rule.

All potential optimal solutions of the CCD scheduling problem can be subdivided into the following three disjunctive cases that are given in Figure 1. For each of them it is assumed that the jobs in B and A are ordered according to the V-shaped property.

Figure 1 The three disjunctive cases of optimal solutions



To represent the problem a permutation is used consisting of the n given jobs of the problem and $|B|_{\max}$ dummy jobs with $p_{n+1} = p_{n+2} = \dots = p_{n+|B|_{\max}} = 0$, i.e., the permutation contains $n + |B|_{\max}$ jobs. It is at most necessary to interpret the first $|B|_{\max} + 1$ positions of this permutation to unequivocally define the represented schedule. The processing times of the jobs are summed up according to the order given by the permutation. The first case occurs if at some point this sum exactly equals the due date. If on the other hand this sum exceeds the due date the current job straddles as depicted in

case two. The third case occurs if the sum of the processing times of the first $|B|_{\max} + 1$ jobs of the permutation is smaller than d (since some of the dummy jobs are among the first positions). This procedure can be summarised as follows: let $[i]$ be the job that occupies the i th position of the permutation:

Problem representation

Step 0: $B = \varnothing, A = \{1, 2, \dots, n + |B|_{\max}\}$

Step 2: while (stop = false) do

If $\text{sum} + P_{[i]} \leq d$ then $\text{sum} = \text{sum} + p_{[i]}, B = B \cup [i]$ and $A = A - [i]$

else $[i]$ is the straddling job,

$A = A - [i], \text{stop} = \text{true}.$

$i = i + 1;$

If $(I > |B|_{\max} + 1)$ or $(\text{sum} = d)$ then $\text{stop} = \text{true}.$

With this procedure an unequivocal assignment of the jobs to the sets A and B is given. Furthermore, it is determined whether a straddling job exists and, if so, which of the jobs this is. The jobs in A and B are ordered according to the V-shaped property. If a straddling job exists the first job of the set B is started in time zero. After finishing the last job of set B the straddling job and afterwards the jobs of the set A are scheduled. If on the other hand no job straddles. The completion of the last job of the set B coincides with the due date and the processing of the jobs in the set A is followed immediately. With the determination of the starting time of the first job and the construction of the sequence the objective function value can be calculated easily. Note that with this problem representation all potential optimal schedules might be represented.

Furthermore, different operators to generate a neighbourhood are possible. A tardy job can be shifted from the set A to the set B , that is from a position $j, j \in |B| + 1, |B| + 2, \dots, |B|_{\max} + n$ of the permutation to a position $k, k \in 1, 2, \dots, |B|_{\max}$. With this operation the jobs occupying the positions $k, k + 1, \dots, j - 1$ are shifted right by one direction. This shifting procedure can be carried out for multiple jobs simultaneously or the tardy job chosen can be shifted to the first position ($k = 1$), etc. From computational experiments, it is decided not to adapt this shifting but rather n interchange procedure. The neighbourhood $N(S)$ used throughout the paper is as follows: Two jobs are selected at random from the sets A and B and are interchanged if at least one of them is not a dummy job. A tardy job at a position $j, j \in |B| + 1, |B| + 2, \dots, |B|_{\max} + n$ is exchanged with an early job at a position $k, k \in 1, 2, \dots, |B|$ of the permutation.

5 Elements of SS

The solution approach that we have developed for the SMSP consists of an adaptation of SS. The SS is an instance of the so-called evolutionary method, which is not based solely on randomisation as the main mechanism for searching. It constructs solutions by

combining others by means of strategic designs that exploit the knowledge on the problem at hand. The goal of these procedures is to enable a solution procedure based on the combined elements to yield better solutions than one based on the original elements.

In common with other evolutionary methods, SS operates with a population of solutions, rather than with a single solution at a time, and employs procedures for combining these solutions to create new ones. The meaning of ‘combining’, and the motivation for carrying it out, has a rather special origin and character in the SS setting. One of the distinguishing features of this approach is its intimate association with the TS meta-heuristic, and hence its adoption of the principle that search can benefit by incorporating special forms of adaptive memory. More about the origin and multiple applications of SS can be found in Glover (1998), Glover et al. (2000) and Laguna et al. (1998). The basic steps involved in the static scatter search are:

- Step 1 use the diversification generator to generate diverse trial solutions from the seed solution(s)
- Step 2 use the improvement method to create one or more enhanced trial solutions
- Step 3 with these initial solutions update the reference set (RefSet)
- Step 4 repeat
 - 4.1 generate subsets of the RefSet
 - 4.2 combine these subsets and obtain new solutions
 - 4.3 use the improvement method to create a more enhanced trial solution
 - 4.4 continue to maintain and update the reference set until Refset is stable (no new solutions are included)
- Step 5 if iterations (Steps 1–4) elapse without improvement stop else return to step 1.

5.1 A diversification generation method

Diversification is to generate a collection of diverse trial solutions, using one or more arbitrary trial solutions (or seed solutions) as an input. This method is suggested by Glover (1998) which generates diversified permutations in a systematic way without reference to the objective function. Assume that a given trial permutation (T) used as a seed is $T = (1, 2, \dots, n)$. Define the subsequence $T(h:s)$, where s is a positive integer between 1 and h , to be given by $T(h:s) = (s, s + h, s + 2h, \dots, s + rh)$, where r is the largest nonnegative integer such that $s + rh \leq n$. Then define the permutation $T(h)$, for $h \leq n$, to be $T(h) = [T(h:h), T(h:h-1), \dots, T(h:1)]$.

Illustration:

Suppose T is given by

$$T = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,)$$

If we choose $h = 5$, then

$$T(5:5) = (5, 10, 15)$$

$$T(5:4) = (4, 9, 14)$$

$$T(5:3) = (3,8,13)$$

$$T(5:2) = (2,7,12)$$

$$T(5:1) = (1,6,11,16)$$

to give:

$$T(5) = (5, 10, 15, 4, 9, 14, 3, 8, 13, 2, 7, 12, 1, 6, 11, 16)$$

In this illustration, ‘h’ is allowed to take the two values closest to the square root of ‘n’. These values are interesting based on the fact that, when ‘h’ equals the square root of ‘n’, the minimum relative separation of each element from each other element in the new permutation is maximum, compared to the relative separation of exactly one in the permutation T. In general, for the goal of generating a diverse set of permutations, preferable values for ‘h’ range from 1 to $n/2$.

5.2 Improvement method

The goal of improvement is to transform a trial solution into one or more enhanced trial solutions. Neither the input nor the output solutions are required to be feasible, though the output solutions are typically feasible. If the input trial solution is not improved as a result of the application of this method, the ‘enhanced’ solution is considered to be the same as the input solution.

The improvement method is based on the neighbourhood search developed for the linear ordering problem (LOP) in Laguna et al. (1998). Four greedy local search procedures for improvement have been compared (Feldmann, 1999) and the greedy procedure found to be the most effective, has been used in this paper. Insertions are used as the primary mechanism to move from one solution to another in our improvement method. We define *MOVE* (p_j, i) to consist of deleting p_j from its current position j in p to be inserted in position i (i.e., between the current elements p_{i-1} and p_i if $i < j$ and between the current elements p_i and p_{i+1} if $i > j$). This operation results in the ordering p' as follows:

$$p' = \begin{cases} (p_1, \dots, p_{i-1}, p_j, p_i, \dots, p_{j-1}, p_{j+1}, \dots, p_\pi) & \text{for } i < j \\ (p_1, \dots, p_{j-1}, p_{j+1}, \dots, p_i, p_j, p_{i+1}, \dots, p_\pi) & \text{for } i > j \end{cases}$$

Since the local search method is context independent, the only available mechanism for computing the move value is submitting p' for evaluation and comparing its objective value with the objective value of p .

In permutation problems, we accumulate in *FreqIns* (i, j), the number of times that element i has been inserted in position j improving the current solution. Then, given an element i , we compute the move value at the position j where the value of *FreqIns* (i, j) is maximum. We consider that these positions around it are desirable positions for inserting element i .

5.3 The reference set update method

The reference set update method accompanies each application of the improvement method, and update operation consists of maintaining the record of the ‘b’ best solutions

found, where value of ‘b’ is stated as a chosen constant, but may readily be allowed to vary.

The RefSet is a collection of ‘b’ solutions that are used to generate new solutions by way of applying a solution combination method. The construction of the initial reference set in step 3 starts with the selection of the best b/2 solutions from T. These solutions are added to RefSet and deleted from T. The minimum distance from each improved solution in RefSet to the solutions in RefSet is computed. Then, the solution with the maximum of these minimum distances is selected. This solution is added to RefSet and deleted from T and the minimum distances are updated. This process is repeated b/2 times. The resulting reference set has b/2 high quality solutions and b/2 diverse solutions. The distance between the two sequences $p = (p_1, p_2, p_3, \dots, p_n)$ and $q = (q_1, q_2, q_3, \dots, q_n)$ is given by:

$$d(p, q) = \sum_{i=1}^n |p_i - q_i|$$

where $d(p, q)$ = number of times p_{i+1} does not immediately follow p_i in q , for $i = 1, \dots, n-1$.

5.4 Subset generation

A *subset generation method* operates on the reference set, to produce a subset of its solutions as a basis for creating combined solutions. In such situations, we seek a procedure that generates subsets X of RefSet that have useful properties, while avoiding the duplication of subsets previously generated. Our approach for doing this is organised to generate four different collections of subsets of RefSet, which we refer to as sub-set type = 1, 2, 3 and 4.

Subset type	all 2-element subsets
Subset type	2: 3-element subsets derived from the 2-element subsets by augmenting each 2-element subset to include the best solution not in this subset.
Subset type	3: 4-element subsets derived from the 3-element subsets by augmenting each 3-element subset to include the best solutions not in this subset.
Subset type	4: the subsets consisting of the best i elements, for $i = 5$ to b .

The total number of subsets that satisfy the preceding stipulations is usually quite manageable. For example, if $b_{\max} = 10$ there are 45 different 2-elements subsets for subset type = 1, and the collections for subset type = 2 and 3 each contain a bit less than 45 additional subsets. All together, subset type = 1 to 4 would generate approximately 130 distinct subsets.

5.5 Combination method

A *solution combination method* is to transform a given subset of solutions produced by the subset generation method into one or more combined solutions. In this method, the two reference solutions vote for their incipient element to be included in the first still unassigned position of the trial solution. If both solutions vote for the same element, the

element is assigned. If the reference solutions vote for different elements, but these elements occupy the same position in both reference permutations, then the element from the permutation with the better objective function is chosen. Finally, if the elements are different and occupy different positions, then the one in the lower position is selected.

The Figure 2 shows the clear-cut idea of implementation of the SS method for obtaining the optimum solution.

Figure 2 SS method

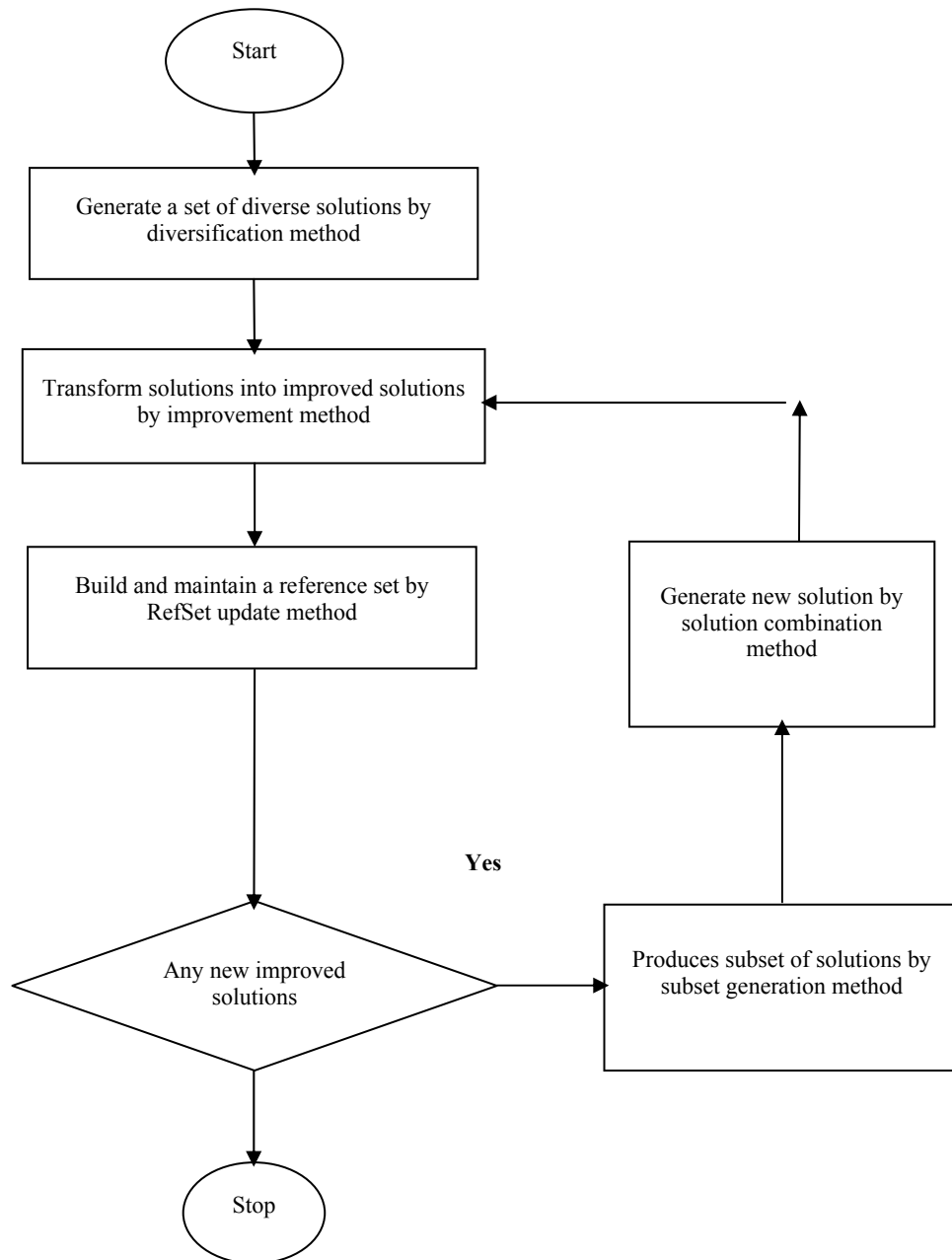


Table 1 Comparison of SS with best objective values for CDD problems

S. no	UB (f^b)	Best-FB (f^*)	SS	PRD (f^b)	PRD (f^*)	UB (f^b)	Best-FB (f^*)	SS	PRD (f^b)	PRD (f^*)
	10 jobs $h = 0.2$					10 jobs $h = 0.4$				
1	2,009		1,936	3.64		1,057		1,025	3.03	
2	1,125		1,042	7.38		615		615	0	
3	1,731		1,586	8.38		931		917	1.51	
4	2,392		2,139	10.58		1,251		1,230	1.68	
5	1,220		1,187	2.71		661		630	4.69	
6	1,623		1,521	6.29		908		908	0	
7	2,269		2,170	4.37		1,374		1,374	0	
8	1,724		1,724	2.82		1,104		1,024	7.25	
9	1,792		1,577	12		876		892	-1.83	
10	1,934		1,869	3.37		1,173		1,138	2.99	
<i>Average</i>				6.54					1.93	
	20 jobs $h = 0.2$					20 jobs $h = 0.4$				
1	4,431	4,394	4,406	0.56	-0.28	3,066	3,066	3,070	-0.13	-0.14
2	8,567	8,430	8,462	1.23	-0.38	4,897	4,847	4,866	0.63	-0.40
3	6,331	6,210	6,277	0.85	-1.08	3,883	3,838	3,892	-0.23	-1.41
4	9,478	9,188	9,215	2.77	-0.30	5,122	5,118	5,150	-0.55	-0.63
5	4,340	4,215	4,245	2.19	-0.72	2,571	2,495	2,527	1.71	-1.29
6	6,766	6,527	6,616	2.22	-1.37	3,601	3,582	3,584	0.47	-0.06
7	11,101	10,455	10,485	5.55	-0.29	6,357	6,238	6,282	1.18	-0.71
8	4,203	3,920	3,946	6.11	-0.67	2,151	2,145	2,157	-0.28	-0.42
9	3,530	3,465	3,498	0.91	-0.96	2,097	2,096	2,154	-2.72	-2.77
10	5,545	4,979	5,042	9.07	-1.27	3,192	2,925	2,098	34.27	28.28
<i>Average</i>				3.15	-0.73				3.44	2.05

Notes: f^* = improved CDD benchmarks
 f^b = CDD Benchmarks

Source: Feldmann and Biskup (2003) and Biskup and Feldmann (2001)

Table 1 Comparison of SS with best objective values for CDD problems (continued)

S. no	UB	Best-FB	SS	PRD	PRD	UB	Best-FB	SS	PRD	PRD
	(f^b)	(f^*)		(f^b)	(f^*)	(f^b)	(f^*)		(f^b)	(f^*)
<i>50 jobs</i>										
<i>h = 0.2</i>										
1	42,363	40,697	40,705	3.91	-0.02	24,868	23,792	24,463	1.63	-2.83
2	33,637	30,613	31,043	7.71	-1.41	19,279	17,907	18,376	4.68	-2.62
3	37,641	34,612	34,895	7.30	-0.82	21,353	20,522	20,787	2.65	-1.30
4	30,166	27,755	28,314	6.14	-2.02	17,495	16,657	16,969	3.01	-1.88
5	32,604	32,307	32,354	0.77	-0.15	18,441	18,007	18,332	0.59	-1.81
6	36,920	34,969	35,114	4.89	-0.41	21,497	20,385	21,078	1.95	-3.40
7	44,277	43,134	43,656	1.40	-1.22	23,883	23,038	23,570	1.31	-2.31
8	46,065	43,839	43,966	4.56	-0.29	25,402	24,895	24,966	1.72	-0.29
9	36,397	34,228	34,546	5.09	-0.93	21,929	19,984	21,473	2.08	-7.46
10	35,797	32,958	33,102	7.53	-0.44	20,048	19,167	19,890	0.79	-3.78
<i>Average</i>				<i>4.93</i>	<i>-0.77</i>				<i>2.04</i>	<i>-2.77</i>
<i>100 jobs</i>										
<i>h = 0.2</i>										
1	156,103	145,516	145,592	6.73	-0.05	89,588	85,884	87,912	1.87	-2.36
2	132,605	124,927	124,991	5.74	-0.05	74,854	72,983	73,011	2.46	-0.04
3	137,463	129,835	129,855	5.53	-0.02	85,363	79,619	81,918	4.04	-2.89
4	137,265	129,584	129,685	5.52	-0.08	87,830	79,413	86,917	1.04	-9.45
5	136,761	124,417	129,506	5.30	-4.09	76,424	71,316	74,820	2.10	-4.91
6	151,938	139,193	140,013	7.85	-0.59	86,724	77,797	84,916	2.08	-9.15
7	141,613	135,089	137,014	3.25	-1.42	79,854	78,244	78,133	2.16	0.14
8	168,086	160,147	161,156	4.12	-0.63	95,361	94,368	94,299	1.11	0.07
9	125,153	116,522	116,610	6.83	-0.08	73,605	69,458	72,816	1.07	-4.83
10	124,446	118,911	118,913	4.45	0.00	72,399	71,850	72,337	0.09	-0.68
<i>Average</i>				<i>5.53</i>	<i>-0.70</i>				<i>1.80</i>	<i>-3.41</i>

Notes: f^* = improved CDD benchmarks
 f^b = CDD Benchmarks

Source: Feldmann and Biskup (2003) and Biskup and Feldmann (2001)

6 Computational results

One of the difficulties faced by the researchers in scheduling is to compare their methodologies with those of other researchers. If the standard set of test problems is

accessible, the performances of different algorithms can be compared on exactly the same set of test problems. For this reason, to clearly indicate the solution capability of our approach, CCD benchmark problems are chosen for which upper bounds on the optimal objective function value exist, see Biskup and Feldmann (2001). These are problems with $n = 10, 20, 50, 100, 200, 500$ and $1,000$ jobs. For each of these problem sizes, 10 different instances with two different values for the due date, $d = \left[h \sum_{i=1}^n p_i c \right]$ with $h = 0.2$ and 0.4 are taken into account. However, the upper bounds for the benchmark problems of Biskup and Feldmann (2001) can be found in the OR Library under <http://mscmga.ms.ic.ac.uk/info.html>.

CCD problems are solved with $n = 10, 20, 50$ and 100 jobs. For each problem size, ten different instances and two different due dates [$h = 0.2$ and 0.4] are considered for each instance. Hence totally 80 problems are solved whose results are compared with the best result (Best-FB) for each of the benchmark problem of the three meta-heuristic technique used by Feldmann and Biskup (2003). The results of the three meta-heuristic techniques used by Feldmann and Biskup (2003) for the same bench mark problems is available in OR Library or it can be down loaded from the website <http://www.wiwi.uni-bielefeld.de/~kistner/Bounds.html>).

The SS technique is compared with the techniques of Feldmann and Biskup (2003). Results are summarised in Table 1 for the benchmark problems attempted by these authors in order to provide a more balanced analysis.

The percentage relative difference (PRD) is used as the measure to compare the results presented here and those of other methods.

$$\text{PRD} = (\text{Best-FB} - \text{SS}_{\text{SOL}}) / \text{Best-FB}$$

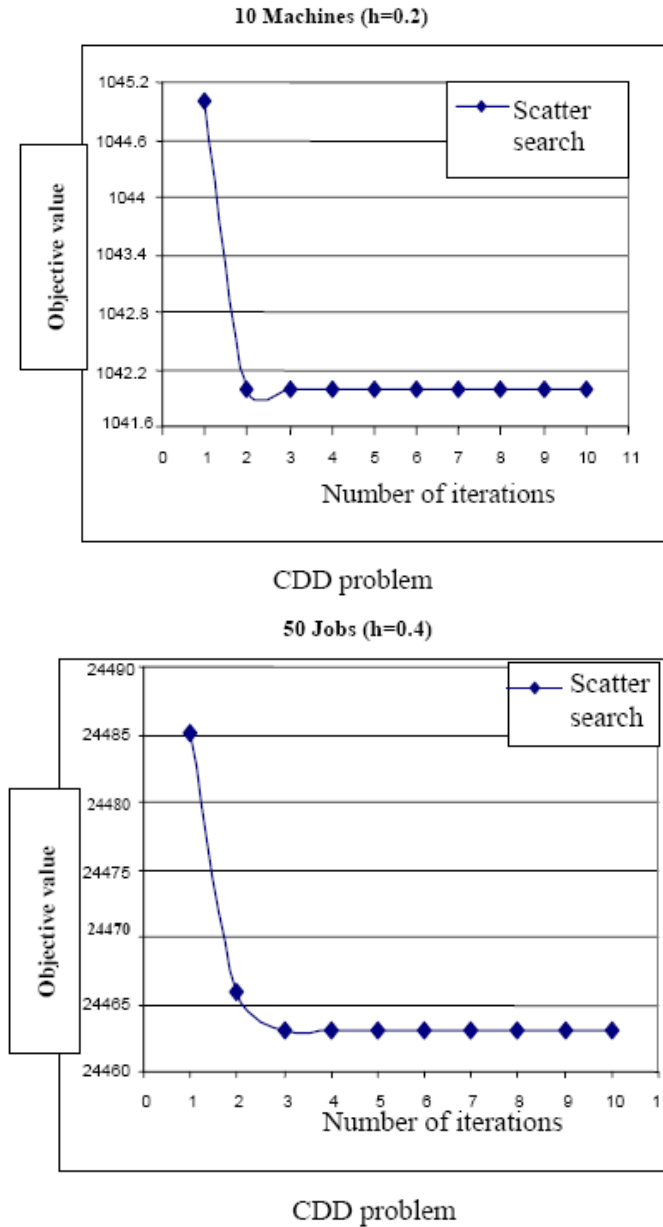
Here Best-FB is the best known solution among the three meta-heuristics of Feldmann and Biskup (2003) and SS_{SOL} is the best solution found by the scatter search method.

Table 1 clearly shows that only a few improved objective values are achieved by the SS method and most of the solutions are as same as the F-Best value in comparison to Martin Feldmann and Dirk Biskup (2003) problem instances. In Table 1, the average PRD for 20 jobs instances is positive, showing that SS is better than the best of the available techniques. Though instances for 50 and 100 jobs shows negative slopes it is comparatively negligible. On an average, the SS method provides better results as in the case of Biskup and Feldmann (2001). It can also be seen in Table 1 that a few new best solutions have been obtained in this technique. This makes it clearly evident that SS is an efficient method to solve single machine scheduling with CCD problems.

7 Performance test

In order to compare the performance of the proposed SS method for SMSPs, the speed of convergence of optimum value versus number of iteration is presented here. In this view, two different size problems of Biskup and Feldmann (2001) benchmark instances have been selected and run on the computer for ten numbers of iterations. The recorded values of the objective value for all the ten iterations are plotted in the graph. The curves obtained for the instances are shown in Figure 3. One can easily observe the convergence of SS for CCD problems. The optimum results are obtained in less number of iterations.

Figure 3 Convergence analysis of SS for SMS problems (see online version for colours)



8 Discussion and conclusions

The main aim of this research is to explore the potential of SS for scheduling problems of a single machine scheduling with CDD. The inherent weakness of many search procedures is that they often get trapped in a region around some local minima. The ability of the SS is to breakout of such entrapments and achieves better, ideally global

minima, is based on their capacity to provide a suitable mixture of intensification and diversification. SS also provides unifying principles for joining solutions based on generalised path constructions and by utilising strategic designs where as other approaches resort to randomisation. Additional advantages are provided by intensification and diversification mechanisms, which exploit adaptive memory, together with processes that avoid generating duplicate solutions at various stages. This is demonstrated clearly by the improved performance of this metaheuristic in comparison to the methods of Biskup and Feldmann (2001). In fact, SS metaheuristic is capable of achieving best results for the benchmark problems of Biskup and Feldmann (2001). Out of 80 benchmark problems of Biskup and Feldmann (2001), the SS technique gives best solutions for all the 80 problem instances. It emerges that the objective values of the benchmark problems have improved by 3.70% on average. At the same time, the objective values of the SS technique gives almost the same values as the best objective values of the three methods used by Feldmann and Biskup (2003) which is clearly showed in the graphs (Figure 4).

Figure 4 Comparison of objective values in graphical method

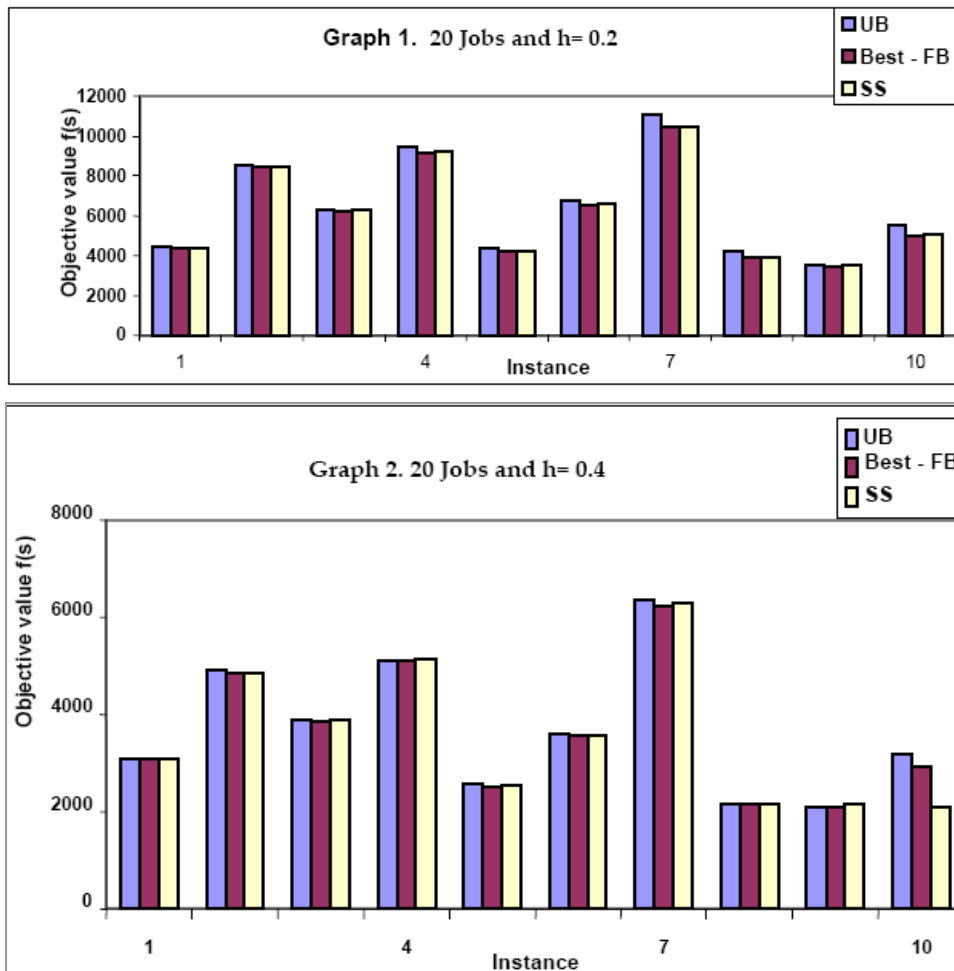
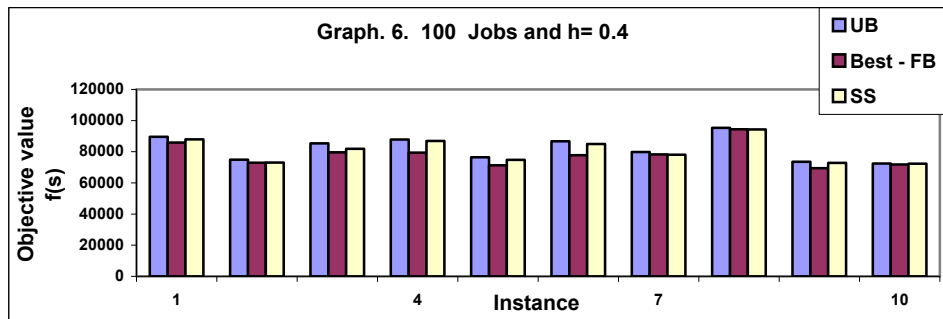
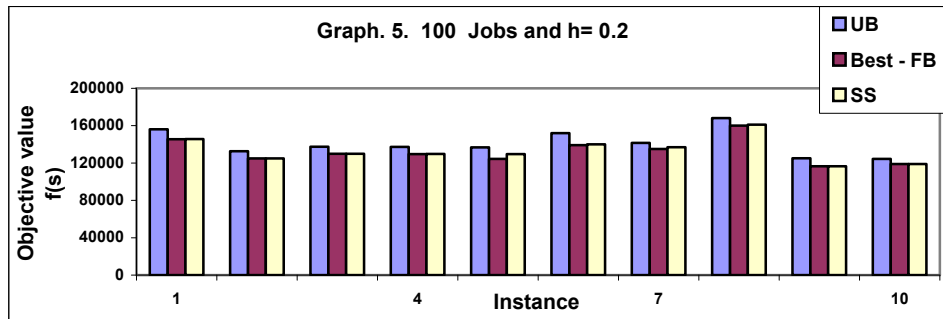
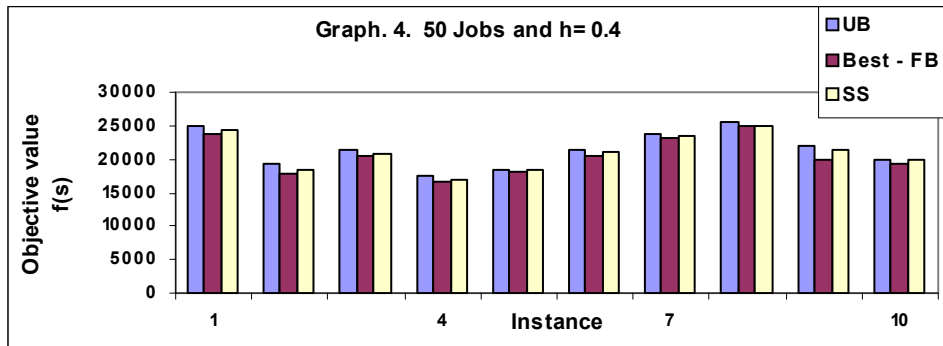
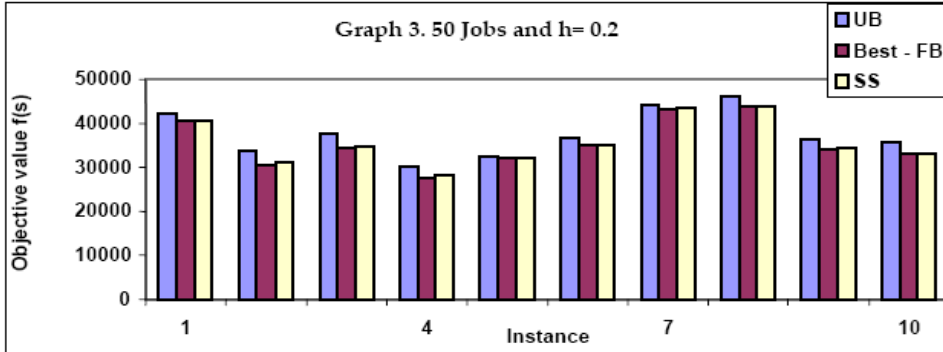


Figure 4 Comparison of objective values in graphical method (continued)



References

- Bagchi, U., Sullivan, R. and Chang, Y. (1987) 'Minimizing mean squared deviation of completion times about a common due date', *Management Science*, Vol. 33, pp.894–906.
- Baker, K. and Scudder, G. (1989) 'On the assignment of optimal due dates', *Journal of the Operational Research Society*, Vol. 40, pp.93–95.
- Baker, K.R. and Scudder, G.D. (1990) 'Sequencing with earliness and tardiness penalties: a review', *Operations Research*, Vol. 38, pp.22–36.
- Biskup, D. and Cheng, T.C.E. (1999) 'Multiple-machine scheduling with earliness, tardiness and completion time penalties', *Computers and Operations Research*, Vol. 26, pp.45–57.
- Biskup, D. and Feldmann, M. (2001) 'Benchmarks for scheduling on a single-machine against restrictive and unrestrictive common due dates', *Computers and Operations Research*, Vol. 28, pp.787–801.
- Cheng, T. and Gupta, M. (1989) 'Survey of scheduling research involving due date determination decisions', *European Journal of Operational Research*, Vol. 38, pp.156–166.
- Cheng, T.C.E. (1987) 'An algorithm for the CON due date determination and sequencing problem', *Computers and Operations Research*, Vol. 14, pp.537–542.
- Cheng, T.C.E. (1990) 'Optimal common due date with limited completion time deviation', *Computer and Operations Research*, Vol. 30, pp.91–96.
- Cheng, T.C.E. and Kahlbacher, H.G. (1991) 'A proof for the longest-job-first policy in one-machine scheduling', *Naval Research Logistics*, Vol. 38, pp.715–720.
- Conway, R.W. (1965) 'Priority dispatching and job lateness in a job shop', *Journal of Industrial Engineering*, Vol. 16, pp.228–237.
- Emmons, H. (1987) 'Scheduling to a common due date on parallel common processors', *Naval Research Logistics Quarterly*, Vol. 34, pp.803–810.
- Feldmann, M. (1999) *Naturalanaloge Verfahren*, Wiesbaden: Gabler, in German.
- Feldmann, M. and Biskup, D. (2003) 'Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches', *Computers & Industrial Engineering*, Vol. 44, pp.307–323.
- Glover, F. (1998) 'A template for scatter search and path relinking', in Hao, J-K, Lutton, E., Ronald, E., Schoenauer, M. and Snyers, D. (Eds): *Artificial Evolution*, Lecture notes in Computer Science, Vol. 1363, pp.13–54, Springer, Berlin.
- Glover, F., Laguna, M. and Marti, R. (2000) 'Fundamentals of scatter search and path relinking', *Control and Cybernetics*, Vol. 39, No. 3, pp.653–684.
- Gordon, V., Proth, J.M. and Chu, C. (2002) 'A survey of the state-of-the-art of common due date assignment and scheduling research', *European Journal of Operational Research*, Vol. 139, No. 1, pp.1–25.
- Hall, N., Kubiak, G.W. and Sethi, S.P. (1991) 'Earliness-tardiness scheduling problems, II: deviation of completion times about a restrictive common due date', *Operations Research*, Vol. 39, pp.847–856.
- Hall, N.G. (1986) 'Single- and multiple-processor models for minimizing completion time variance', *Naval Research Logistics Quarterly*, Vol. 33, pp.49–54.
- Hall, N.G. and Posner, M.E. (1991) 'Earliness-tardiness scheduling problems, I: weighted deviation of completion times about a common due date', *Operations Research*, Vol. 39, pp.836–846.
- Hall, N.G. and Posner, M.E. (1991) 'Earliness-tardiness scheduling problems, I: weighted deviation of completion times about a common due date', *Operations Research*, Vol. 39, pp.836–846.
- Hino, C.M., Ronconi, D.P. and Mendes, A.B. (2005) 'Minimizing earliness and tardiness penalties in a single-machine problem with a common due date', *European Journal of Operational Research*, Vol. 160, No. 1, pp.190–201.
- Hoogeveen, J.A. and van de Velde, S.L. (1991) 'Scheduling around a small common due date', *European Journal of Operational Research*, Vol. 55, pp.237–242.

- James, R.J.W. (1997) 'Using tabu search to solve the common due date early/tardy machine scheduling problem', *Computers and Operations Research*, Vol. 24, pp.199–208.
- Kanet, J.J. (1981) 'Minimizing the average deviation of job completion times about a common due date', *Naval Research Logistics Quarterly*, Vol. 28, pp.643–651.
- Laguna, M., Martí, R. and Campos, V. (1998) 'Intensification and diversification with elite tabu search solutions for the linear ordering problem', to appear in *Computers and Operations Research*.
- Laguna, M., Martí, R. and Campos, V. (1998) 'Intensification and diversification with elite tabu search solutions for the linear ordering problem', to appear in *Computers and Operations Research*.
- Lee, C-Y. and Kim, S.J. (1995) 'Parallel genetic algorithms for the earliness-tardiness job scheduling problem with general penalty weights', *Computers and Industrial Engineering*, Vol. 28, pp.231–243.
- Lee, C-Y., Danusaputro, S.L. and Lin, C-S. (1991) 'Minimizing weighted number of tardy jobs and weighted earliness-tardiness penalties about a common due date', *Computers and Operations Research*, Vol. 18, pp.379–389.
- Noorul Haq, A. and Saravanan, M. (2007) 'A scatter search approach for minimizing earliness and tardiness penalties in single machine scheduling around large restrictive common due windows', *International Journal of Applied Management and Technology*, Vol. 5, No. 2, pp.76–91.
- Noorul Haq, A. and Saravanan, M. et al. (2007) 'A scatter search approach for general flow shop scheduling problem', *International Journal of Advanced Manufacturing Technology*, Vol. 31, pp.731–736.
- Osman, I.H. and Laporte, G. (1996) 'Metaheuristics: a bibliography', *Annals of Operations Research*, Vol. 63, pp.513–625.
- Panwalkar, S.S., Smith, M.L. and Seidmann, A. (1982) 'Common due date assignment to minimize total penalty for the one machine-scheduling problem', *Operations Research*, Vol. 30, pp.391–399.
- Saravanan, M. and Noorul Haq, A. (2008) 'A scatter search method to minimise makespan of cell scheduling problem', *International Journal of Agile Systems and Management*, Vol. 3, Nos. 1/2, pp.18–36.
- Saravanan, M. and Noorul Haq, A. (2008) 'Evaluation of scatter search approach for scheduling optimization of flexible manufacturing systems', *International Journal of Advanced Manufacturing Technology*, Vol. 38, Nos. 9/10, pp.978–986.
- Saravanan, M. and Noorul Haq, A. (2010), 'A scatter search algorithm for scheduling optimisation of job shop problems', *International Journal of Product Development*, Vol. 10, Nos. 1/2/3, pp.259–272.
- Saravanan, M. and Noorul Haq, A. et al. (2008) 'Performance evaluation of scatter search method for permutation flow-shop sequencing problems', *International Journal of Advanced Manufacturing Technology*, Vol. 37, Nos. 11/12, pp.1200–1208.
- Saravanan, M. and Sundararaman, K.A. (2010) 'Ant colony optimisation based heuristic for one-sided time constraint vehicle routing problem', *Int. J. Services, Economics and Management*, Vol. 2, Nos. 3/4, pp.332–349.
- Seyed, M.A. and Mohammad, T.T. (2008) 'Evaluating meta-heuristic algorithms for solving restricted single machine scheduling problems: a comparative analysis', *World Applied Sciences Journal*, Vol. 4, No. 1, pp.75–86.
- Sidney, J.B. (1977) 'Optimal single machine scheduling with earliness and tardiness penalties', *Operations Research*, Vol. 25, pp.62–69.
- Szwarc, W. (1989) 'Single-machine scheduling to minimize absolute deviation of completion times from a common due date', *Naval Research Logistics*, Vol. 36, pp.663–673.
- Wang, J-B. (2006) 'Single machine scheduling with common due date and controllable processing times', *Applied Mathematics and Computation*, Vol. 174, No. 2, 15 March, pp.1245–1254.

- Webster, S. and Gupta, D. (1998) 'A genetic algorithm for scheduling job families on a single machine with arbitrary earliness/tardiness penalties and an unrestricted common due date', *International Journal of Production Research*, Vol. 36, No. 9, pp.2543–2551.
- Weng, M. and Ventura, J. (1994) 'A note on 'single machine scheduling for minimizing total cost with identical, asymmetrical earliness and tardiness penalties'', *International Journal of Production Research*, Vol. 32, pp.2727–2729.
- Ying, K-C. (2008) 'Minimizing earliness-tardiness penalties for common due date single machine scheduling problems by a recovering beam search algorithm', *Computers and Industrial Engineering*, Vol. 55, No. 2, pp.494–502.