

Bi-criteria single machine scheduling problem with a learning effect: Aneja–Nair method to obtain the set of optimal sequences

V. Mani^a, Pei Chann Chang^{b,*}, Shih Hsin Chen^c

^a Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India

^b Department of Information Management, Yuan Ze University, 135, Yuan-Dong Road, Tao-Yuan 32026, Taiwan ROC

^c Department of Electronic Commerce Management, Nanhua University, 32, Chungkeng, Dalin, Chiayi 62248, Taiwan ROC

ARTICLE INFO

Article history:

Received 24 October 2007

Received in revised form 12 November 2008

Accepted 12 January 2009

Keywords:

Single machine scheduling

Bi-criteria problem

Learning effect

Non-dominated solution

ABSTRACT

In this paper, we consider the bi-criteria single machine scheduling problem of n jobs with a learning effect. The two objectives considered are the total completion time (TC) and total absolute differences in completion times ($TADC$). The objective is to find a sequence that performs well with respect to both the objectives: the total completion time and the total absolute differences in completion times. In an earlier study, a method of solving bi-criteria transportation problem is presented. In this paper, we use the methodology of solving bi-criteria transportation problem, to our bi-criteria single machine scheduling problem with a learning effect, and obtain the set of optimal sequences. Numerical examples are presented for illustrating the applicability and ease of understanding.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

During the past fifty years, the single machine scheduling problem has been studied by many researchers. A good introduction to sequencing and scheduling, and also presenting various issues related to single machine scheduling is [1]. In single machine scheduling, various objectives such as mean flow time, mean tardiness, maximum flow time, maximum tardiness, number of tardy jobs, weighted mean of earliness and tardiness are considered. The problem is to find a sequence of jobs that minimizes the considered objective. In single machine scheduling, the processing time of a job is assumed to be a constant. A well-known concept in management science literature is “learning effect” first discovered in [2]. Because of this learning effect, the processing time of a job is not a constant and depends on its position in the sequence. A survey of learning effect is given in [3].

In the context of single machine scheduling, the learning effect was first considered in [4]. The objectives considered in [4] are minimal deviation from a common due date and minimum flow time, and these objectives are studied separately. An assignment problem formulation is presented for each of these objectives, and the solution is obtained by solving the assignment problem. The assignment problem formulation given in [4] with the objective of due-date assignment problem, simultaneous minimization of total completion time and variation of completion times is presented in [5]. The scheduling problem with a learning effect on parallel identical machines is presented in [6]. A two machine flowshop with a learning effect, with the objective of minimizing of total completion time, is studied in [7]. In [7] there is proposed a branch and bound algorithm to solve this problem and a heuristic algorithm is also presented to improve the efficiency of the branch and bound technique. Some important studies considering the learning effect are: [8–12]. The bi-criteria single machine scheduling problem with a learning effect is considered in [13] and the two objectives considered are the total completion

* Corresponding author. Tel.: +886 3 4638800x2305, 2509; fax: +886 3 4638907.

E-mail address: iepchang@saturn.yzu.edu.tw (P.C. Chang).

time and the maximum tardiness. A branch and bound technique is presented for the solution and a heuristic algorithm is proposed to search for optimal and near optimal solutions.

In this paper, we include the learning effect as given in [4]. The processing time of a job depends on its position in the sequence and is given as [4]

$$p_{jl} = p_j l^\alpha. \quad (1)$$

In the above equation, p_j is the normal processing time of job j , and p_{jl} is the processing time of job j if it is in position l of the sequence, and α is the learning index and $\alpha < 0$. From the above Eq. (1), we see that $p_{j1} > p_{j2} > p_{j3} \dots > p_{jn}$. For example, if $p_j = 3$ and $\alpha = -0.515$, then $p_{j1} = 3$, $p_{j2} = 2.0994$, $p_{j3} = 1.7037$, $p_{j4} = 1.4691$, $p_{j5} = 1.3095$, and so on.

Contributions of this paper: In this paper, we consider the bi-criteria single machine scheduling problem of n jobs with a learning effect. The two objectives considered are the total completion time (TC) and total absolute differences in completion times ($TADC$). The objective is to find a sequence that performs well with respect to both the objectives: the total completion time and the total absolute differences in completion times. A method of solving the bi-criteria transportation problem is presented in [14]. In our study, we use the method of solving bi-criteria transportation problem [14], to our bi-criteria single machine scheduling problem with a learning effect, and obtain the set of optimal sequences. This bi-criteria problem was considered in an earlier study [15] without the learning effect. A parametric analysis was presented to obtain the minimum set of optimal sequences in [15]. In that study, first the complete set of optimal solutions (CSOS) is obtained and then, from this CSOS, the minimum set of optimal solutions (MSOS) are obtained. We also present a discussion on the parametric analysis given in [15]. A pseudo-polynomial time dynamic programming algorithm is presented in [16] to solve this bi-criteria problem. We also show that, in our methodology, we can directly obtain the minimum set of optimal solutions (MSOS).

2. Aneja–Nair method of solution

In this section, we outline Aneja–Nair method [14], of solving the bi-criteria transportation problem. The algorithm presented in [14], consists of solving the same transportation problem repeatedly but with different objectives. In each iteration, a new efficient extreme point or the change of direction of search in the objective function is obtained. This algorithm terminates when there is no more efficient extreme point.

$$\begin{aligned} \text{Minimize } z_1 &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{Minimize } z_2 &= \sum_{i=1}^m \sum_{j=1}^n d_{ij} x_{ij} \end{aligned} \quad (2)$$

subject to the constraints

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= a_i, \quad i = 1, 2, \dots, m \\ \sum_{i=1}^m x_{ij} &= b_j, \quad j = 1, 2, \dots, n \\ x_{ij} &\geq 0, \quad \text{forall } (i, j), \end{aligned} \quad (3)$$

where

- c_{ij} is the cost of transporting a unit from source i to destination j ,
- d_{ij} is the deterioration of a unit while transporting from i to j ,
- a_i is the availability at i ,
- b_j is the requirement at j ,
- x_{ij} is the amount transported from i to j .

The algorithm presented in [14] starts with two points in the objective space. These points are $z^{(1)}$ and $z^{(2)}$. Points $z^{(1)}$ and $z^{(2)}$ are obtained by minimizing the first objective and second objective respectively. The value of first and second objective function at point 1 are denoted as $z_1^{(1)}, z_2^{(1)}$. The value of first and second objective function at point 2 are denoted as $z_1^{(2)}, z_2^{(2)}$. Using the values of $z_1^{(1)}, z_2^{(1)}, z_1^{(2)},$ and $z_2^{(2)}$, two new variables namely $a_1^{(1,2)}$, and $a_2^{(1,2)}$ are determined. The value of $a_1^{(1,2)} = |z_2^{(2)} - z_2^{(1)}|$, and $a_2^{(1,2)} = |z_1^{(2)} - z_1^{(1)}|$. The new transportation problem is formulated as

$$\text{Minimize } \sum_{i,j} \left\{ a_1^{(1,2)} * c_{ij} + a_2^{(1,2)} * d_{ij} \right\} x_{ij}. \quad (4)$$

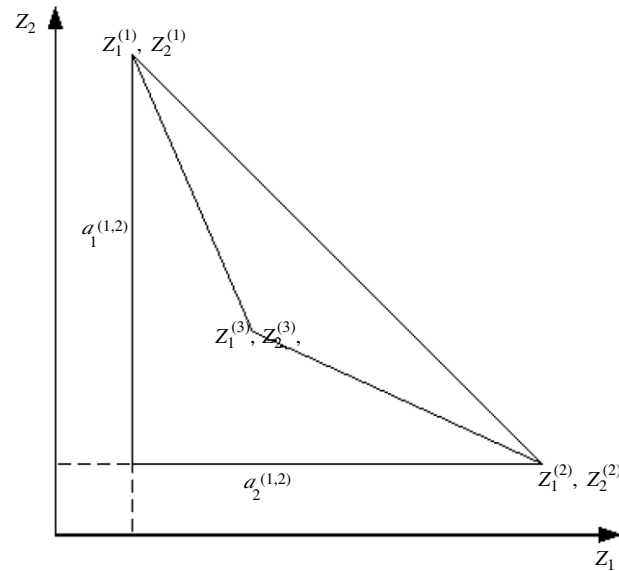


Fig. 1. Pictorial explanation of objective space.

The solution of the above transportation problem gives a new efficient extreme point or change of direction of search. If there are alternate optimal solutions to this problem, choose an optimal solution for which $\sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij}$ is a minimum. Let the new extreme point obtained be point 3. The value of first and second objective function at point 3 are denoted as $z_1^{(3)}, z_2^{(3)}$. This process is repeated by considering points 1 and 3 and also by considering points 2 and 3. If a new efficient extreme point is found from points 1 and 3, the process continues with the new point along with points 1 and 3. Similarly, if a new efficient extreme point is found from points 2 and 3, the process continues with the new point along with points 2 and 3. This algorithm terminates when there is no more efficient extreme point. The proof of validity of this algorithm is given in [14].

Pictorial explanation: In order to understand the algorithm given in [14], we give below a pictorial representation. In that study [14], the objective space is considered instead of the decision space for finding the extreme points on the non-dominated set. The objective space for the bi-criteria transportation problem can be thought of as an $x - y$ plane, in which x -axis represents the first objective function value (z_1) and y -axis represents the first objective function value (z_2). Any point in this plane gives the value of first objective function (z_1) and the value of second objective function (z_2) for a given transportation problem. This is shown in Fig. 1. In this Fig. 1, the points $z_1^{(1)}, z_2^{(1)}$ and $z_1^{(2)}, z_2^{(2)}$ are shown. Also, the value of $a_1^{(1,2)}, a_2^{(1,2)}$ and the third point $z_1^{(3)}, z_2^{(3)}$ obtained by solving (4) are shown. This process is repeated by considering points 1 and 3 and also by considering points 2 and 3.

3. Problem formulation and preliminary analysis

We consider the single machine scheduling problem with a learning effect. A set of n independent jobs is to be processed on a continuously available single machine. The machine can process only one job at a time and job preemption and inserting idle times are not permitted. Each job has a normal processing time p_j , ($j = 1, 2, \dots, n$) if it is at the first position in the sequence. The sequence is the order in which the jobs are processed on the machine. The jobs are numbered according to shortest normal processing time rule, i.e., $p_1 \leq p_2 \leq \dots \leq p_n$. The two objectives considered are the total completion time (TC) and total absolute differences in completion times ($TADC$). TC and $TADC$ for a given sequence σ are

$$TC = \sum_{j=1}^n C_j$$

$$TADC = \sum_{i=1}^n \sum_{j=i}^n |C_i - C_j| \tag{5}$$

where C_j is the completion time of job j in the given sequence. The single machine scheduling problem with a learning effect, is to find the sequence of jobs (σ) that simultaneously minimizes TC and $TADC$ are shown in Eqs. (6) and (7), respectively.

$$f_{TC}(\sigma) = \sum_{r=1}^n (n - r + 1)r^\alpha p_{[r]} = \sum_{r=1}^n w_r^{1,\alpha} p_{[r]} \tag{6}$$

Table 1
Matching algorithm for minimization of TC: $\alpha = -0.152$.

Position- r	1	2	3	4
$w_r^{1,\alpha}$	4.0000	2.7000	1.6924	0.8100
Sequence*	1	2	3	4

Table 2
Matching algorithm for minimization of TADC: $\alpha = -0.152$.

Position- r	1	2	3	4
$w_r^{2,\alpha}$	0.0000	2.7000	3.3848	2.4300
Sequence*	4	2	1	3

and

$$f_{TADC}(\sigma) = \sum_{r=1}^n (r-1)(n-r+1)r^\alpha p_{[r]} = \sum_{r=1}^n w_r^{2,\alpha} p_{[r]}. \tag{7}$$

In the above equations, $p_{[r]}$ is the normal processing time of job in position r . The quantities $(n-r+1)r^\alpha$ and $(r-1)(n-r+1)r^\alpha$ in the above equations, are known as positional weights. We can also see if $\alpha = 0$ Eqs. (6) and (7) reduces to the problem presented in [15]. The optimal sequence for any one of the objectives can be obtained by using a matching algorithm [15,17].

Numerical example: We now give a small numerical example to show how the matching algorithm is used to obtain the optimal schedule. Consider the 4 job instance of the problem given in [15]. The normal processing time of these jobs are $p_1 = 1, p_2 = 2, p_3 = 3$, and $p_4 = 4$. Let the value of learning rate $\alpha = -0.152$.

Consider only the first objective i.e., TC: The problem is to find the sequence of jobs (σ) that minimizes

$$f_{TC}(\sigma) = \sum_{r=1}^n (n-r+1)r^\alpha p_{[r]} \tag{8}$$

$$f_{TC}(\sigma) = 4 * 1^\alpha p_{[1]} + 3 * 2^\alpha p_{[2]} + 2 * 3^\alpha p_{[3]} + 1 * 4^\alpha p_{[4]}. \tag{9}$$

The optimal sequence obtained for the above problem using the matching algorithm is {1 2 3 4}. The positional weights obtained from Eq. (6) are given in Table 1. The details of obtaining the optimal sequence using the matching algorithm is given in Appendix. The value of TC obtained for this sequence {1 2 3 4} is 17.7173. The value of TADC obtained for this sequence {1 2 3 4} is 25.2746.

Consider only the second objective; i.e., TADC: The problem is to find the sequence of jobs (σ) that minimizes

$$f_{TADC}(\sigma) = \sum_{r=1}^n (r-1)(n-r+1)r^\alpha p_{[r]} \tag{10}$$

$$f_{TADC}(\sigma) = 3 * 2^\alpha p_{[2]} + 4 * 3^\alpha p_{[3]} + 3 * 4^\alpha p_{[4]}. \tag{11}$$

The optimal sequence obtained for the above problem using the matching algorithm is {4 2 1 3}. The positional weights obtained from Eq. (10) are given in Table 2. The details of obtaining the optimal sequence using the matching algorithm is given in Appendix. The value of TADC obtained for this sequence {4 2 1 3} is 16.0749. The value of TC obtained for this sequence {4 2 1 3} is 25.5224. Our objective is to find the sequence of jobs that performs well with respect to both the objectives TC and TADC. In the next section, we describe the methodology of obtaining the set of sequences that performs well with respect to both the objectives TC and TADC, using the method given in [14].

4. Aneja–Nair method to bi-criteria single machine scheduling

Our interest lies in finding a sequence that minimizes both TC and TADC. Now, we will show how the bi-criteria problem in single machine scheduling with a learning effect can be solved to obtain the minimum set of optimal schedules using Aneja–Nair algorithm [14].

First, we will show the one to one correspondence of variables between bi-criteria transportation problem and bi-criteria single machine scheduling problem with a learning effect. The one to one correspondence between the two problems is given in Table 3. Once this is known, then we can apply the Aneja–Nair method to obtain the minimum set of optimal sequences.

The algorithm [14] for our problem is that of solving the single-machine scheduling problem (with a learning effect) repeatedly but with different objectives. In each iteration, we obtain an optimal sequence (efficient point). In this manner, at any iteration, we are solving the single machine scheduling (with a learning effect) problem with a single (but different) objective function, using the matching algorithm given in [17]. We present a numerical example for illustration.

Table 3
One to one correspondence between the two problems.

Bi-criteria transportation problem	Bi-criteria single machine scheduling problem
z_1	TC first objective
z_2	TADC second objective
c_{ij}	$(n - r + 1) * r^\alpha = w_r^{1,\alpha}$
d_{ij}	$(r - 1)(n - r + 1) * r^\alpha = w_r^{2,\alpha}$
Extreme point	optimal sequence

Table 4
Matching algorithm for minimization of (12): $\alpha = -0.152$.

Position-r	1	2	3	4
$w_r^{1,\alpha}$	36.7988	45.9131	41.9887	26.4985
Sequence*	3	1	2	4

Table 5
Matching algorithm for minimization of (13): $\alpha = -0.152$.

Position-r	1	2	3	4
$w_r^{1,\alpha}$	24.3396	26.1698	22.5093	13.6953
Sequence*	2	1	3	4

Numerical example: We now use the same 4 job problem instance given in [15], with a learning effect ($\alpha = -0.152$). We show how to apply Aneja and Nair method [14] and obtain the minimum set of optimal schedules. The normal processing time of these jobs are $p_1 = 1, p_2 = 2, p_3 = 3,$ and $p_4 = 4$. The value of learning rate $\alpha = -0.152$.

Following Aneja and Nair [14], we first obtain point 1 in the objective space. This is obtained by minimizing z_1 the first objective; i.e., the total completion time (TC). The positional weights and the optimal sequence obtained are shown in Table 1. The optimal sequence obtained is {1 2 3 4}. The value of z_1 for this sequence is 17.7173. The value of z_2 the second objective; i.e., the total absolute differences in completion times (TADC), is 25.2746. Hence, $z_1^{(1)} = 17.7173,$ and $z_2^{(1)} = 25.2746$.

We now obtain point 2 in the objective space. This is obtained by minimizing z_2 the second objective; i.e., the total absolute differences in completion times (TADC). The positional weights and the optimal sequence obtained are shown in Table 2. The optimal sequence obtained is {4 2 1 3}. The values of z_1 and z_2 for this sequence ({4 2 1 3}) are 25.5224 and 16.0749 respectively. Hence, $z_1^{(2)} = 25.5224,$ and $z_2^{(2)} = 16.0749$.

As mentioned in [14], we now use the points 1 and 2, and obtain the values of $a_1^{(1,2)} = |z_2^{(2)} - z_2^{(1)}| = 9.1997,$ and $a_2^{(1,2)} = |z_1^{(2)} - z_1^{(1)}| = 7.8051$. The new single machine scheduling problem is formulated as to find the sequence of jobs (σ) that minimizes

$$f(\sigma) = \sum_{r=1}^n a_1^{(1,2)} * (n - r + 1)r^\alpha p_{[r]} + \sum_{r=1}^n a_2^{(1,2)} (r - 1)(n - r + 1)r^\alpha p_{[r]}. \tag{12}$$

The positional weights are $w_r^{c,\alpha} = a_1^{(1,2)} * (n - r + 1)r^\alpha p_{[r]} + a_2^{(1,2)} (r - 1)(n - r + 1)r^\alpha p_{[r]}$. The positional weights ($w_r^{c,\alpha}$) for the above combined problem are: $w_1^{c,\alpha} = 36.7988, w_2^{c,\alpha} = 45.9131, w_3^{c,\alpha} = 41.9887,$ and $w_4^{c,\alpha} = 26.4985$. The positional weights and the optimal sequence obtained are shown in Table 4. The optimal sequence obtained with these weights is {3 1 2 4}. The values of z_1 and z_2 for this sequence ({3 1 2 4}) are 21.3249 and 19.1897 respectively. We call this point 3 in the objective space and $z_1^{(3)} = 21.3249,$ and $z_2^{(3)} = 19.1897$. We now use the points 1 and 3, and obtain the values of $a_1^{(1,3)} = |z_2^{(3)} - z_2^{(1)}| = 6.0849,$ and $a_2^{(1,3)} = |z_1^{(3)} - z_1^{(1)}| = 3.6079$. The new single machine scheduling problem is formulated as to find the sequence of jobs (σ) that minimizes

$$f(\sigma) = \sum_{r=1}^n a_1^{(1,3)} * (n - r + 1)r^\alpha p_{[r]} + \sum_{r=1}^n a_2^{(1,3)} (r - 1)(n - r + 1)r^\alpha p_{[r]}. \tag{13}$$

The positional weights are $w_r^{c,\alpha} = a_1^{(1,3)} * (n - r + 1)r^\alpha p_{[r]} + a_2^{(1,3)} (r - 1)(n - r + 1)r^\alpha p_{[r]}$. The positional weights ($w_r^{c,\alpha}$) for the above combined problem are: $w_1^{c,\alpha} = 24.3396, w_2^{c,\alpha} = 26.1698, w_3^{c,\alpha} = 22.5093,$ and $w_4^{c,\alpha} = 13.6953$. The positional weights and the optimal sequence obtained are shown in Table 5. The optimal sequence obtained with these weights is {2 1 3 4}. The values of z_1 and z_2 for this sequence ({2 1 3 4}) are 18.2173 and 22.5746 respectively. We call this point 4 in the objective space and $z_1^{(4)} = 18.2173,$ and $z_2^{(4)} = 22.5746$. We now use points 2 and 3, and obtain the values of $a_1^{(2,3)} = |z_2^{(3)} - z_2^{(2)}| = 3.1148,$ and $a_2^{(2,3)} = |z_1^{(3)} - z_1^{(2)}| = 4.1975$. The new single machine scheduling problem is formulated so as to find the sequence of jobs (σ) that minimizes

$$f(\sigma) = \sum_{r=1}^n a_1^{(2,3)} * (n - r + 1)r^\alpha p_{[r]} + \sum_{r=1}^n a_2^{(2,3)} (r - 1)(n - r + 1)r^\alpha p_{[r]}. \tag{14}$$

Table 6
Matching algorithm for minimization of (14): $\alpha = -0.152$.

Position- r	1	2	3	4
$w_r^{1,\alpha}$	12.4592	19.7433	19.4794	12.7230
Sequence*	4	1	2	3

The positional weights are $w_r^{c,\alpha} = a_1^{(2,3)} * (n - r + 1)r^\alpha p_{[r]} + a_2^{(2,3)} (r - 1)(n - r + 1)r^\alpha p_{[r]}$. The positional weights ($w_r^{c,\alpha}$) for the above combined problem are: $w_1^{c,\alpha} = 12.4592$, $w_2^{c,\alpha} = 19.7433$, $w_3^{c,\alpha} = 19.4794$, and $w_4^{c,\alpha} = 12.7230$. The positional weights and the optimal sequence obtained are shown in Table 6. The optimal sequence is obtained with these weights is {4 1 2 3}. The values of z_1 and z_2 for this sequence ({4 1 2 3}) are 24.5149 and 16.7597 respectively. We call this point 5 in the objective space and $z_1^{(5)} = 24.5149$, and $z_2^{(5)} = 16.7597$. When we use the points 1 and 4, we obtain the same optimal sequence {1 2 3 4}. When we use the points 3 and 4, we obtain the same optimal sequence obtained is {3 1 2 4}. When we use the points 3 and 5, we obtain the same optimal sequence {3 1 2 4}. When we use the points 2 and 5, we obtain the same optimal sequence {4 1 2 3}. There are no other optimal sequences and so the algorithm terminates.

Based on the above, the minimum set of optimal sequences to this problem is: {1 2 3 4}, {4 2 1 3}, {3 1 2 4}, {2 1 3 4}, and {4 1 2 3}.

It is shown in [15] that the cardinality of the set MSOS is n , when the learning effect is not considered; i.e., $\alpha = 0$. We can see from this example, that the cardinality of the set MSOS is not n , when the learning effect is included; i.e., $\alpha \neq 0$.

5. Discussions

This bi-criteria single machine scheduling problem without a learning effect ($\alpha = 0$) is studied in Bagchi [15]. In that study, a parametric analysis is carried out. In the parametric analysis, the objective is to find the sequence (σ) that minimizes

$$f(\sigma) = \delta * \sum_{r=1}^n TC + (1 - \delta) * \sum_{r=1}^n TADC \tag{15}$$

$$f(\sigma) = \delta * \sum_{r=1}^n (n - r + 1)p_{[r]} + (1 - \delta) * \sum_{r=1}^n (r - 1)(n - r + 1)p_{[r]}. \tag{16}$$

The value of δ is restricted to the open interval (0,1). For any given value of δ , the optimal sequence can be obtained by using the matching procedure. In that study [15], two sets of sequences namely Complete Set of Optimal Schedules (CSOS), and Minimum Set of Optimal Schedules (MSOS), are defined. The set CSOS has the property that for any value of δ from (0,1), all optimal solutions to (15) are members of this set. The set MSOS has the property that for any value of δ from (0,1), at least one and almost of the optimal solutions to (15) are members of this set. So MSOS is a subset of CSOS.

It is shown in [15] that there are $(n - 1)$ distinct values of δ given by

$$\delta_i = \frac{n - i}{n - i + 1} \quad i = 1, 2, \dots, (n - 1). \tag{17}$$

A fast algorithm for obtaining the CSOS is given in [15]. A dynamic programming approach is presented in [16]. In that study, a condition on δ_i is given by considering three adjacent extreme points. The minimal set of optimal sequences are called δ -efficient sequences in [16]. It was shown in Bagchi [15] that the cardinality of the set MSOS is n , and an $O(n^2)$ algorithm is presented to obtain the set MSOS. We present a numerical example to show the sets CSOS and MSOS.

Numerical example: We now use the same 4 job problem given in Bagchi [15], without a learning effect ($\alpha = 0$). The normal processing time of these jobs are $p_1 = 1, p_2 = 2, p_3 = 3$, and $p_4 = 4$. For this 4 job bi-criteria problem, there are 3 distinct values of δ . They are: $\delta_1 = 0.5, \delta_2 = 2/3$, and $\delta_3 = 0.75$.

We consider the value of $\delta = 0.5$. The objective is to find the sequence (σ) that minimizes

$$f(\sigma) = 0.5 * \sum_{r=1}^n TC + 0.5 * \sum_{r=1}^n TADC \tag{18}$$

$$f(\sigma) = 0.5 * \sum_{r=1}^n (n - r + 1)p_{[r]} + 0.5 * \sum_{r=1}^n (r - 1)(n - r + 1)p_{[r]}. \tag{19}$$

The positional weights (w_r^c) for the above combined problem are: $w_1^c = 2.0000, w_2^c = 3.0000, w_3^c = 3.0000$, and $w_4^c = 2.0000$. We see that the positional weights $w_1^c = w_4^c$, and $w_2^c = w_3^c$. Because of this, we obtain four sequences that are optimal. The four optimal sequences are: {3 1 2 4}, {3 2 1 4}, {4 1 2 3}, and {4 2 1 3}. For all these four sequences, the value of $\delta * \sum_{r=1}^n TC + (1 - \delta) * \sum_{r=1}^n TADC$ is the same and the value is 23.

We consider the value of $\delta = 2/3$. We conduct a similar analysis as done above. We obtain two sequences that are optimal. The two optimal sequences are: {2 1 3 4} and {3 1 2 4}. For the two sequences, the value of $\delta * \sum_{r=1}^n TC + (1 - \delta) * \sum_{r=1}^n TADC$ is the same and the value is 23.

We consider the value of $\delta = 3/4$. We conduct a similar analysis as done above. We obtain two sequences that are optimal. The two optimal sequences are: {1 2 3 4} and {2 1 3 4}. For the two sequences, the value of $\delta * \sum_{r=1}^n TC + (1 - \delta) * \sum_{r=1}^n TADC$ is the same and the value is 22.50.

The complete set of optimal schedules (CSOS) and the minimum set of optimal schedules (MSOS) are:

CSOS : {{1 2 3 4}, {2 1 3 4}, {3 1 2 4}, {3 2 1 4}, {4 1 2 3}, {4 2 1 3}}.

MSOS : {{1 2 3 4}, {2 1 3 4}, {3 1 2 4}, {4 2 1 3}}.

Aneja–Nair method: We now show how the Aneja–Nair method obtains the MSOS for the above example, without a learning effect ($\alpha = 0$). When $\alpha = 0$, we use the key idea given in [14]. The key idea is that when there is more than one optimal solution, choose an optimal solution for which z_1 is minimum.

Following Aneja and Nair [14], we first obtain point 1 in the objective space. This is obtained by minimizing z_1 the first objective; i.e., the total completion time (TC). The positional weights and the optimal sequence obtained is {1 2 3 4}. The value of z_1 for this sequence is 20.0000. The value of z_2 the second objective; i.e., the total absolute differences in completion times (TADC), is 30.0000. Hence, $z_1^{(1)} = 20.0000$, and $z_2^{(1)} = 30.0000$.

We now obtain point 2 in the objective space. This is obtained by minimizing z_2 the second objective; i.e., the total absolute differences in completion times (TADC). The positional weights are: $w_1^2 = 0.0000$, $w_2^2 = 3.0000$, $w_3^2 = 4.0000$, and $w_4^2 = 3.0000$. We see that the positional weights $w_2^2 = w_4^2$. Because of this, we obtain two sequences that are optimal. The two optimal sequences are: {4 2 1 3} and {4 3 1 2}. It is given in Aneja and Nair [14] that when there are more than one optimal solution, choose an optimal solution for which z_1 is minimum. The value of z_1 for the sequence {4 2 1 3} is 27, and the value of z_1 for the sequence {4 3 1 2} is 29. Hence, we choose the sequence {4 2 1 3}. The values of z_1 and z_2 for this sequence ({4 2 1 3}) are 27.0000 and 19.0000 respectively. Hence, $z_1^{(2)} = 27.0000$, and $z_2^{(2)} = 19.0000$.

As mentioned in [14], we now use the points 1 and 2, and obtain the values of $a_1^{(1,2)} = |z_2^{(2)} - z_2^{(1)}| = 11.0000$, and $a_2^{(1,2)} = |z_1^{(2)} - z_1^{(1)}| = 7.0000$. The new single machine scheduling problem is formulated so as to find the sequence of jobs (σ) that minimizes

$$f(\sigma) = \sum_{r=1}^n a_1^{(1,2)} * (n - r + 1)p_{[r]} + \sum_{r=1}^n a_2^{(1,2)}(r - 1)(n - r + 1)p_{[r]}. \tag{20}$$

The positional weights (w_r^c) for the above combined problem are: $w_1^c = 44.0000$, $w_2^c = 54.0000$, $w_3^c = 50.0000$, and $w_4^c = 32.0000$. The optimal sequence obtained with these weights is {3 1 2 4}. The values of z_1 and z_2 for this sequence ({3 1 2 4}) are 23.0000 and 23.0000, respectively. We call this point 3 in the objective space and $z_1^{(3)} = 23.0000$, and $z_2^{(3)} = 23.0000$.

We now use the points 1 and 3, and obtain the values of $a_1^{(1,3)} = |z_2^{(3)} - z_2^{(1)}| = 7.0000$, and $a_2^{(1,3)} = |z_1^{(3)} - z_1^{(1)}| = 3.0000$. The new single machine scheduling problem is formulated so as to find the sequence of jobs (σ) that minimizes

$$f(\sigma) = \sum_{r=1}^n a_1^{(1,3)} * (n - r + 1)p_{[r]} + \sum_{r=1}^n a_2^{(1,3)}(r - 1)(n - r + 1)p_{[r]}. \tag{21}$$

The positional weights (w_r^c) for the above combined problem are: $w_1^c = 28.0000$, $w_2^c = 30.0000$, $w_3^c = 26.0000$, and $w_4^c = 16.0000$. The optimal sequence obtained with these weights is {2 1 3 4}. The values of z_1 and z_2 for this sequence ({2 1 3 4}) are 21.0000 and 27.0000 respectively. We call this point 4 in the objective space and $z_1^{(4)} = 21.0000$, and $z_2^{(4)} = 27.0000$.

We now use the points 2 and 3, and obtain the values of $a_1^{(2,3)} = |z_2^{(3)} - z_2^{(2)}| = 4.0000$, and $a_2^{(2,3)} = |z_1^{(3)} - z_1^{(2)}| = 4.0000$. The new single machine scheduling problem is formulated so as to find the sequence of jobs (σ) that minimizes

$$f(\sigma) = \sum_{r=1}^n a_1^{(2,3)} * (n - r + 1)p_{[r]} + \sum_{r=1}^n a_2^{(2,3)}(r - 1)(n - r + 1)p_{[r]}. \tag{22}$$

The positional weights (w_r^c) for the above combined problem are: $w_1^c = 16.0000$, $w_2^c = 24.0000$, $w_3^c = 24.0000$, and $w_4^c = 16.0000$. We see that the positional weights $w_1^c = w_4^c$, and $w_2^c = w_3^c$. Because of this, we obtain four sequences that are optimal. The four optimal sequences are: {3 1 2 4}, {3 2 1 4}, {4 1 2 3}, and {4 2 1 3}. Of these four sequences, the sequence with a minimum value of z_1 is chosen. The sequence with a minimum z_1 is the sequence {4 2 1 3}.

In this manner, we obtain the minimum set of optimal schedules (MSOS) and the sequences are:

MSOS : {{1 2 3 4}, {2 1 3 4}, {3 1 2 4}, {4 2 1 3}}.

It is shown by [15] that there are $(n - 1)$ distinct values of δ given by the Eq. (17). Using these distinct values, it is shown in [15] that the cardinality of the set MSOS is n , when the learning effect is not considered; i.e., $\alpha = 0$. When the learning effect is included i.e., $\alpha \neq 0$, the Eq. (17) is not true and hence the cardinality of the set MSOS is not n . This fact is shown in this example.

The effect of learning in scheduling problems has attracted many researchers and some of the recent studies are given in [18–23].

6. Conclusions

In a manufacturing system, workers are involved in doing the same job or activity repeatedly. Because of repeating the same job or activity, the workers start learning more about the job or activity. Because of the learning, the time to complete the job or activity starts decreasing, which is known as “learning effect.” Because of this learning effect, the processing time of a job is not a constant and depends on its position in the sequence. In this paper, we considered the bi-criteria single machine scheduling problem of n jobs with a learning effect. The two objectives considered are the total completion time (TC) and total absolute differences in completion times ($TADC$). We used Aneja–Nair [14] method to find a sequence that performs well with respect to both the objectives: the total completion time and the total absolute differences in completion times. In an earlier study [15] this bi-criteria problem is considered without the learning effect, and a parametric analysis is presented to obtain the minimum set of optimal sequences. In this study, we used the Aneja–Nair method [14] of solving bi-criteria transportation problem, to obtain the set of optimal sequences, to our bi-criteria single machine scheduling problem with a learning effect. A discussion is presented on the parametric analysis given in [15], and the methodology used in our study. We have also presented numerical examples for ease of understanding.

Acknowledgments

The authors would like to thank one of the reviewers for his/her suggestions which were useful. The authors would like to thank Editor-in-Chief Prof. Rodin for his encouragement.

Appendix

Numerical example: We now give a small numerical example to show how the matching algorithm is used to obtain the optimal sequence. Consider the 4 job problem given in [15]. The normal processing time of these jobs are $p_1 = 1$, $p_2 = 2$, $p_3 = 3$, and $p_4 = 4$. Let the value of learning rate $\alpha = -0.152$.

We first consider only the first objective i.e., TC . The problem is to find the sequence of jobs (σ) that minimizes

$$f_{TC}(\sigma) = \sum_{r=1}^n (n-r+1)r^\alpha p_{[r]} \quad (23)$$

$$f_{TC}(\sigma) = 4 * 1^\alpha p_{[1]} + 3 * 2^\alpha p_{[2]} + 2 * 3^\alpha p_{[3]} + 1 * 4^\alpha p_{[4]}. \quad (24)$$

The positional weights obtained from Eq. (6) are $w_1^{1,\alpha} = 4.0000$, $w_2^{1,\alpha} = 2.7000$, $w_3^{1,\alpha} = 1.6924$, and $w_4^{1,\alpha} = 0.8100$. In the matching algorithm, the optimal sequence is obtained by matching the positional weights in descending order with jobs in ascending order of their normal processing times. The positional weights and the optimal sequence obtained are shown in Table 1. The optimal sequence obtained is {1 2 3 4}, when we consider only the first (TC) objective.

The value of TC for this sequence {1 2 3 4} is obtained as follows:

$$z_1 = \sum_{r=1}^n w_r^{1,\alpha} p_{[r]}. \quad (25)$$

In the above equation $p_{[r]}$ is the normal processing time of job in position r . We have job 1 in position 1, job 2 in position 2, job 3 in position 3, and job 4 in position 4. Hence, z_1 is

$$\begin{aligned} z_1 &= w_1^{1,\alpha} p_1 + w_2^{1,\alpha} p_2 + w_3^{1,\alpha} p_3 + w_4^{1,\alpha} p_4 \\ z_1 &= 4.0000 * 1 + 2.7000 * 2 + 1.6924 * 3 + 0.8100 * 4 = 17.7173. \end{aligned} \quad (26)$$

The value of $TADC$ for this sequence {1 2 3 4} is obtained as follows:

$$z_2 = \sum_{r=1}^n w_r^{2,\alpha} p_{[r]}. \quad (27)$$

We have job 1 in position 1, job 2 in position 2, job 3 in position 3, and job 4 in position 4. Hence, z_2 is

$$\begin{aligned} z_2 &= w_1^{2,\alpha} p_1 + w_2^{2,\alpha} p_2 + w_3^{2,\alpha} p_3 + w_4^{2,\alpha} p_4 \\ z_2 &= 0.0000 * 1 + 2.7000 * 2 + 3.3848 * 3 + 2.4300 * 4 = 25.2746. \end{aligned} \quad (28)$$

We now consider only the second objective; i.e., $TADC$. The problem is to find the sequence of jobs (σ) that minimizes

$$f_{TADC}(\sigma) = \sum_{r=1}^n (r-1)(n-r+1)r^\alpha p_{[r]} \quad (29)$$

$$f_{TADC}(\sigma) = 3 * 2^\alpha p_{[2]} + 4 * 3^\alpha p_{[3]} + 3 * 4^\alpha p_{[4]}. \quad (30)$$

The positional weights obtained from Eq. (7) are $w_1^{2,\alpha} = 0.0000$, $w_2^{2,\alpha} = 2.7000$, $w_3^{2,\alpha} = 3.3848$, and $w_4^{2,\alpha} = 2.4300$. In the matching algorithm, the optimal sequence is obtained by matching the positional weights in descending order with jobs in ascending order of their normal processing times. The positional weights and the optimal sequence obtained are shown in Table 2. The optimal sequence obtained is {4 2 1 3}, when we consider only the second (*TADC*) objective.

The value of *TC* for this sequence {4 2 1 3} is obtained as follows:

$$z_1 = \sum_{r=1}^n w_r^{1,\alpha} p_{[r]}. \quad (31)$$

We have job 4 in position 1, job 2 in position 2, job 1 in position 3, and job 3 in position 4. Hence, z_1 is

$$\begin{aligned} z_1 &= w_1^{1,\alpha} p_4 + w_2^{1,\alpha} p_2 + w_3^{1,\alpha} p_1 + w_4^{1,\alpha} p_3 \\ z_1 &= 4.0000 * 4 + 2.7000 * 2 + 1.6924 * 1 + 0.8100 * 3 = 25.5224. \end{aligned} \quad (32)$$

The value of *TADC* for this sequence {4 2 1 3} is obtained as follows:

$$z_2 = \sum_{r=1}^n w_r^{2,\alpha} p_{[r]}. \quad (33)$$

We have job 4 in position 1, job 2 in position 2, job 1 in position 3, and job 3 in position 4. Hence, z_2 is

$$\begin{aligned} z_2 &= w_1^{2,\alpha} p_4 + w_2^{2,\alpha} p_2 + w_3^{2,\alpha} p_1 + w_4^{2,\alpha} p_3 \\ z_2 &= 0.0000 * 4 + 2.7000 * 2 + 3.3848 * 1 + 2.4300 * 3 = 16.0749. \end{aligned} \quad (34)$$

References

- [1] K.R. Baker, Introduction to Sequencing and Scheduling, John Wiley & Sons, New York, 1974.
- [2] T.P. Wright, Factors affecting the cost of airplanes, Journal of Aeronautical Science 3 (1936) 122–128.
- [3] L.E. Yelle, The learning curve: Historical review and comprehensive survey, Decision Science 10 (1979) 302–328.
- [4] D. Biskup, Single-machine scheduling with learning considerations, European Journal of Operational Research 115 (1999) 173–178.
- [5] G. Mosheiov, Scheduling problems with a learning effect, European Journal of Operational Research 132 (2001) 687–693.
- [6] G. Mosheiov, Parallel machine scheduling with a learning effect, Journal of the Operational Research Society 52 (2001) 1165–1169.
- [7] W.C. Lee, C.C. Wu, Minimizing total completion time in a two-machine flowshop with a learning effect, International Journal of Production Economics 88 (2004) 85–93.
- [8] D. Biskup, D. Simons, Common due date scheduling with autonomous and induced learning, European Journal of Operational Research 159 (2004) 606–616.
- [9] W.H. Kuo, D.L. Yang, Minimizing the makespan in a single machine scheduling problem with a time based learning effect, Information Processing Letters 97 (2006) 64–67.
- [10] G. Mosheiov, J.B. Sidney, Note on scheduling with general learning curves to minimize the number of tardy jobs, Journal of the Operational Research Society 56 (2005) 110–112.
- [11] A. Bachman, A. Janiak, Scheduling jobs with position-dependent processing times, Journal of the Operational Research Society 55 (2004) 257–264.
- [12] J.B. Wang, Z.Q. Xia, Flowshop scheduling with a learning effect, Journal of the Operational Research Society 56 (2005) 1325–1330.
- [13] W.C. Lee, C.C. Wu, H.J. Sung, A bi-criteria single-machine scheduling problem with learning considerations, Acta Informatica 40 (2004) 303–315.
- [14] Y.P. Aneja, K.P.K. Nair, Bicriterion transportation problem, Management Science 25 (1979) 73–78.
- [15] U. Bagchi, Simultaneous minimization of mean and variation of flow time and waiting time in single machine systems, Operations Research 37 (1989) 118–125.
- [16] P. De, J.B. Ghosh, C.E. Wells, On the minimization of completion time variance with a bi-criteria extension, Operations Research 40 (1992) 1148–1155.
- [17] S.S. Panwalkar, M.L. Smith, A. Seidmann, Common due-date assignment to minimize total penalty for the one machine scheduling problem, Operations Research 30 (1982) 391–399.
- [18] D. Biskup, A state-of-the-art review on scheduling with learning effects, European Journal of Operational Research 188 (2008) 315–329.
- [19] A. Janiak, R. Rudek, The learning effect: Getting to the core of the problem, Information Processing Letters 103 (2007) 183–187.
- [20] A. Janiak, R. Rudek, A new approach to the learning effect: Beyond the learning curve restrictions, Computers & Operations Research 35 (2008) 3727–3736.
- [21] C. Koulamas, G.J. Kyparisis, Single-machine and two-machine flow-shop scheduling with general learning functions, European Journal of Operational Research 178 (2007) 402–407.
- [22] J.B. Wang, C.T.D. Ng, T.C.E. Cheng, L.L. Liu, Single-machine scheduling with a time-dependent learning effect, International Journal of Production Economics 111 (2008) 802–811.
- [23] C.C. Wu, W.C. Lee, Single-machine scheduling problems with a learning effect, Applied Mathematical Modelling 32 (2008) 1191–1197.