International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:9, No:10, 2015

Solution Approaches for Some Scheduling Problems with Learning Effect and Job Dependent Delivery Times

M. Duran Toksarı, B. Uçarkuş

***Abstract*—**In this paper, we propose two algorithms to optimally solve makespan and total completion time scheduling problems with learning effect and job dependent delivery times in a single machine environment. The delivery time is the extra time to eliminate adverse effect between the main processing and delivery to the customer. In this paper, we introduce the job dependent delivery times for some single machine scheduling problems with position dependent learning effect, which are makespan are total completion. The results with respect to two algorithms proposed for solving of the each problem are compared with LINGO solutions for 50-jobs, 100-jobs and 150-jobs problems. The proposed algorithms can find the same results in shorter time.

***Keywords*—**Delivery times, learning effect, makespan, scheduling, total completion time.

I. Introduction

IN some manufacturing stages, the processing time of a job is exposed to external effects. In some electronic manufacturing processes, an electronic component in the electromagnetic field requires an extra time in order to eliminate any adverse effects. This additional time period was first introduced by Koulamas and Kyparisis [1] and entitled as the *past sequence dependent (PSD) delivery times*. Koulamas and Kyparisis [1] tried this new concept with several single machine scheduling problems, makespan minimization problem, maximum lateness, maximum tardiness and minimization the number of tardy jobs. They [1] reduced the problems, with the exception of the makespan minimization problem, to their original formulation (without the *PSD* delivery times). Recently, many researchers [1]-[8] have worked a variety of scheduling problems which consider *PSD* delivery times on both single-machine and/or multi-machine settings. In the scheduling literature, some researchers have worked scheduling problems *PSD* under some effects such as learning effect and/or deterioration. The joint feature of these all studies is the common of *PSD* normalizing coefficient for all jobs. Shen and Wu [6] consider polynomial time procedures to solve single machine *PSD* delivery times scheduling with general position dependent and time dependent learning effects. Liu [5] proposes the minimizing of total absolute deviation of job completion times, the total load on all parallel machines and the total completion time with the *PSD* delivery time and learning effect. Liu et al. [4] present polynomial algorithms for the problem with the total workload, the total completion time, the total absolute differences in completion times with past-sequence-dependent delivery times and a deterioration effect. Yang et al. [8] focus on a set of single machine problems with *PSD* delivery times and both effects (learning and deterioration effect) simultaneously. In this paper, we introduce two single-machine scheduling problems, makespan and total completion times, with job dependent *PSD* delivery times under general position dependent learning effect. The basic difference between this paper and other studies is that the *PSD* normalizing coefficient should not be common for all jobs.

M. D. Toksarı is with the Erciyes University, Engineering Faculty, Industrial Engineering Department, Kayseri, Turkey (phone: +90 352 4374901/32457; fax: +90 352 4375784; e-mail: dtoksari@erciyes.edu.tr).

B. Uçarkuş, was with Erciyes University, Engineering Faculty, Industrial Engineering Department, Kayseri, Turkey (e-mail: berrinucarkus@erciyes.edu.tr).

II. Problem Description

In the classical scheduling theory, job processing times are assumed as a constant. However, workers of many real life production systems are exposed to a process of learning [11]-[13]. The repetition of similar operations induces a process of specialization and workers acquire better skills. Mosheiov [9] introduced the common terminology for this phenomenon which is 'learning effect'. Biskup [10] was the first to investigate the learning effect in the context of scheduling. He assumed that the processing time of a job decreases depending on a function of the number of jobs previously processed on the same machine setting. Biskup [10] considered the following model as actual processing time

$$p_{j[r]} = \left(p_j\right) r^{a} \tag{1}$$

where $p_{j[r]}$ is actual processing time of job *J* scheduled in position *r* and p_j is basic processing time. a $\left(a < 0\right)$ is the learning index. Moreover, assumptions in [1], the processing of job $J_{j[r]}$ must be followed the job based *PSD* delivery time $q_{j[r]}$, which can be formulated as

$$q_{j[r]} = \gamma_j W_{j[r]} = \gamma_j \sum_{i=1}^{r-1} p_{j[i]} \qquad \left(j = 1,\ldots,n\right) \tag{2}$$

where $\gamma_j \geq 0$ is the normalizing coefficient of a job and $W_{j[r]}$ is the waiting time of job $J_{j[r]}$. $W_{j[1]} = 0$, since all jobs in a single machine setting with a continuously available machine time are available for processing at time zero.

This paper presents the minimization of two scheduling objectives: makespan $C_{\max} = \max\{C_j \mid j = 1,2,...n\}$ and the total completion time $\sum C_j$. We denote all problems using three field notation scheme $\alpha|\beta|\gamma$ [14].

III. The Makespan and Total Completion Time Scheduling Problems with Job Based *PSD* Delivery Times and Learning Effect

We assume that there are given *n* jobs and single machine, and each machine can handle one job at a time and preemption is not allowed. In the scheduling literature, both makespan and total completion time under general position dependent learning effect $(1|LE|C_{\max}$ and $1|LE|\sum C)$ can be solved optimally by sequencing jobs in non-decreasing order of their processing times (*SPT* rule) (see Theorem 1 and Theorem 2) [9]. Furthermore, these scheduling problems under *PSD* delivery times (with the common of *PSD* normalizing coefficient) and general position dependent learning effect $(1|LE, q_{psd}|C_{\max}$ and $1|LE, q_{psd}|\sum C)$ can be solved optimally by *SPT* rule [15]. In this paper, in order to solve makespan with job based *PSD* delivery times and position dependent learning effect, we propose an algorithm as follows

Theorem 1. The problem $1|p_{[j]}|C_{\max}$ can be solved optimally by sequencing jobs in non-decreasing order their processing times (SPT rule).

Proof. The proof is presented in [9].

Theorem 2. The problem $1|p_{[j]}|\sum C_j$ can be solved optimally by sequencing jobs in non-decreasing order their processing times (SPT rule).

Proof.

The proof is presented in [9].

The Proposed Algorithm

For all jobs $(j = 1,...,n)$

Step1. Assign as job scheduled in the last position $(J_{j[n]})$

Step2. Find the sequencing remained jobs in non-decreasing order of p_j

Step3. Update the best $C_{\max}$ $(or \sum C)$.

To evaluate the performance of the proposed algorithm, a computational experiment was conducted. The proposed algorithm was coded in Visual Studio 2010 C# (see Appendix I for makespan problem), and these problems were modeled with LINGO 8 software to find optimal solution (see Appendix II for makespan problem). The computational experiments were run by computer with 2 Duo 2.66 GHz processor and 4.0 GB RAM. The normal processing time p_j were generated from a random uniform distribution, $p_j \sim U(1,50)$. The values for delivery times were generated from a random uniform distribution, $\gamma_j \sim U(0,1)$. The proposed algorithm for 50-jobs, 100-jobs and 150-jobs problems was evaluated versus the optimal solution obtained by LINGO, and each set was run 50 times. Results show that the proposed algorithm finds sooner the same results with LINGO. Tables I and II show the average solution times of the proposed algorithm and LINGO for three sets (50-jobs, 100-jobs and 150-jobs) of makespan and total completion time scheduling problems, respectively.

TABLE I
THE AVERAGE SOLUTION TIMES FOR MAKESPAN SCHEDULING PROBLEM

Problem	LINGO Average Running Time (second)	The Proposed Algorithm Average Running Time (second)
100-jobs	20.09	0.028
200-jobs	448.42	0.065
300-jobs	59489.27	0.133

TABLE II
THE AVERAGE SOLUTION TIMES FOR TOTAL COMPLETION TIME SCHEDULING PROBLEM

Problem	LINGO Average Running Time (second)	The Proposed Algorithm Average Running Time (second)
100-jobs	22.14	0.031
200-jobs	527.08	0.068
300-jobs	71698.13	0.128

The results evinces that the proposed algorithm finds sooner the same results with LINGO.

IV. Conclusion

This paper introduced single-machine problems with job dependent delivery times and learning effect. The delivery time was assumed to be proportional to the job waiting time. We investigated the objectives consist of minimizing the makespan and the total completion time when the past sequence dependent delivery time based on the job under position dependent learning effect. We propose an algorithm to solve these problems, and the problems were modeled using LINGO software to evaluate the performance of developed algorithm. The results clearly show that the proposed algorithm finds the same results sooner with LINGO.

Appendix

I. The modeling of problem using LINGO software

```
MODEL:
        n=100;  ! Problem size;
      LE=0.8;  ! Learning effect;

SETS:
     JOB / 1.. 100/;
  POSITION / 1.. 100/;
    LINK(JOB,POSITION):
          Z;
    LINK1(POSITION):
```

```
        PROCESSING_TIME,
        COMPLETION_TIME,
                S,
              TOTAL;
      LINK2(JOB):
       T,
      PROCESSING_TIME1;
  ENDSETS
  DATA:
       PROCESSING_TIME1=    @FILE('d:\\T.txt');    !    Basic
processing time;
       T= @FILE('d:\\G.txt');   ! Gama;
       @TEXT('d:\\S.txt')=@WRITE(@OBJBND());
       @TEXT('d:\\T1.txt')=@WRITE(@TIME());
   ENDDATA
  MIN = COMPLETION_TIME(n);
  @FOR( POSITION( R):
     PROCESSING_TIME(R)=@SUM(                    JOB(
I):PROCESSING_TIME1(I)*Z(R,I)*@POW(R,(@LOG10(LE)/@L
OG10(2)))) ;
  );
  @FOR( POSITION( R):
  TOTAL(R)=@SUM(POSITION(J)|J#LT#R:PROCESSING_TIM
E(J));
  );
  @FOR( POSITION( R):
     S(R)=TOTAL(R)*@SUM( JOB( I):T(I)*Z(R,I));
  );
     @FOR( POSITION( R):
  COMPLETION_TIME(R)=(S(R)+TOTAL(R))+(PROCESSING_
TIME(R));
  );
  @FOR( JOB( I): @SUM( POSITION( R): Z( R,I))=1;
  );
  @FOR( POSITION( R): @SUM( JOB( I): Z( R,I))=1;
  );
  @FOR( LINK: @BIN( Z));
  END
```

II. C# codes for the proposed algorithms

```
private void button1_Click(object sender, EventArgs e)
        {
            Random A = new Random();
            textBox1.Text = "";
            textBox3.Text = "";
            listBox1.Items.Clear();
            double[] p = new
double[Convert.ToInt32(textBox2.Text)];
            double[] p1 = new
double[Convert.ToInt32(textBox2.Text)];
            double[] gama = new
double[Convert.ToInt32(textBox2.Text)];
            double[] gama1 = new
double[Convert.ToInt32(textBox2.Text)];
            double[] opts = new
double[Convert.ToInt32(textBox2.Text)];
            double[] optgama = new
double[Convert.ToInt32(textBox2.Text)];
            string s1 = "";
            string s2 = "";
            for (int i = 0; i < p.Length; i++)
            {
               p[i] = A.Next(10,50);
               s1 = s1 + " " + p[i];
               gama[i] = A.Next(1, 100);
               gama[i] = gama[i]/100;
               s2 = s2 + " " + gama[i];
            }
            StreamWriter yaz = new StreamWriter("d:\\T.txt");
            yaz.WriteLine(s1);
            yaz.Close();
            s2 = s2.Replace(',', '.');
            StreamWriter yaz1 = new StreamWriter("d:\\G.txt");
            yaz1.WriteLine(s2);
            yaz1.Close();


            double deg=0;
            double cmax = 0;
            double makespan = 1000000000000;
            for (int i = 0; i < p.Length; i++)
            {
               for (int j = i + 1; j < p.Length;j++ )
                  if (p[i] > p[j])
                  {
                     deg = p[i];
                     p[i] = p[j];
                     p[j] = deg;
                     deg = gama[i];
                     gama[i] = gama[j];
                     gama[j] = deg;
                  }
            }
            string s = "";
            for (int i = 0; i < p.Length-1; i++)
            {
               s = "";
               cmax = 0;
               p1[p.Length - 1] = p[i];
               gama1[p.Length - 1] = gama[i];
               for (int j = 0; j < i; j++)
               {
                  p1[j] = p[j];
                  gama1[j] = gama[j];
                  s = s + "  " + p1[j];
               }
               for (int k = i; k < p.Length-1; k++)
               {
                  p1[k] = p[k+1];
                  gama1[k] = gama[k + 1];
                  s = s + "  " + p1[k];
               }

               for (int l = 0; l < p.Length; l++)
               {
                  cmax = cmax + (p1[l] * Math.Pow(l + 1,
(Math.Log10(0.8) / Math.Log10(2))));
               }
               cmax = cmax + (gama1[p.Length - 1] * (cmax-
(p1[p.Length-1] * Math.Pow(p.Length, (Math.Log10(0.8) /
Math.Log10(2))))));
               cmax = Math.Round(cmax,2);
               if (makespan > cmax)
               {
                  makespan = cmax;
```

```
            for (int m = 0; m < p.Length; m++)
            {
               opts[m] = p1[m];
               optgama[m] = gama1[m];
            }
         }
         s = s + "  " + p1[p.Length - 1] + "   = " +cmax ;
         listBox1.Items.Add(s);
      }
      for (int i = 0; i < p.Length; i++)
      {
         textBox1.Text = textBox1.Text + "  " + opts[i];
         textBox3.Text = textBox3.Text + "  " + optgama[i];
      }
      label1.Text = Convert.ToString(makespan);
   }
```

ACKNOWLEDGMENT

This research was supported by Scientific Research Fund of Erciyes University under the contract no: FBA-2014-5397.

REFERENCES

[1] C. Koulamas, G.J. Kyparisis, Single machine problems with past sequence dependent delivery times, International Journal of Production Economics 126(2), 264-266, 2010.
[2] M. Liu, F. Zheng, C. Chu, Y. Xu, New results on single machine scheduling with past sequence dependent delivery times, Theoretical Computer Science 438(22), 55-61, 2012.
[3] M. Liu, F. Zheng, C. Chu, Y. Xu, Single machine scheduling with past sequence dependent delivery times and release times, Information Processing Letters 112(21), 835-838, 2012.
[4] M. Liu, S. Wang, C. Chu, Scheduling deteriorating jobs with past sequence dependent delivery times, International Journal of Production Economics 144(2), 418-421, 2013.
[5] M. Liu, Parallel machine scheduling with past sequence dependent delivery times and learning effect, Applied Mathematical Modelling 37(23), 9630-9633, 2013.
[6] L. Shen, Y.B. Wu, Single machine past sequence dependent delivery times scheduling with general position dependent and time dependent learning effects, Applied Mathematical Modelling 37, 5444-5451, 2013.
[7] S.J. Yang, D.L. Yang, Single machine scheduling problems with past sequence dependent delivery times and position dependent processing times, Journal of Operational Research Society 63(11), 1508-1514, 2012.
[8] S.J. Yang, J.Y. Guo, H.T. Lee, D.L. Yang, Single machine scheduling problems with past sequence dependent delivery times and deterioration and learning effects simultaneously, International Journal of Innovative Computing, Information and Control 9(10), 3981-3989, 2013.
[9] G. Mosheiov, Scheduling problems with a learning effect, European Journal of Operational Research 132, 687– 693, 2001.
[10] D. Biskup, Single-machine scheduling with learning considerations. European Journal of Operational Research 115, 173–178, 1999.
[11] M.D. Toksari, E. Guner, The common due date Early/tardy scheduling problem on a parallel machine under the effects of time dependent learning and linear/ nonlinear deterioration, Expert Systems with Applications 37(1) (2010) 92-112.
[12] M.D. Toksari, E. Guner, Parallel machine scheduling problem to minimize the earliness/tardiness costs with learning effect and deteriorating jobs, Journal of Intelligent Manufacturing 21(6) (2010) 843-851.
[13] M.D. Toksari, A branch and bound algorithm for minimizing makespan on a single machine with unequal release times under learning effect and deteriorating jobs, Computers and Opeations Research 38(9) (2011) 1361-1365.
[14] R.L. Graham, J.K. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, Annals of Discrete Mathematics 5, 287-326, 1979.
[15] S.-J. Yang, C.-J. Hsu, T.-R. Chang and D.-L. Yang, Single-machine scheduling with past-sequence dependent delivery times and learning effect, Journal of the Chinese Institute of Industrial Engineers, vol.28, pp.247-255, 2011.