Scheduling Jobs with Values Exponentially Deteriorating over Time in a Job Shop Environment

Cheng-Hsiang Liu

Abstract—**This study focuses on solving a special kind of job shop scheduling problem (JSP), where the job value is exponentially deteriorating over time. The current study attempted to find out whether the expected benefits of Lot Streaming (LS) can be found in solving the JSP with the objective of maximizing the total value of the jobs. LS is a process of splitting jobs into smaller sub-jobs such that successive operations can be overlapped. Since the studied scheduling problem is a complex problem, this study proposed an efficient technique comprised of a genetic algorithm (GA) for lot streaming and simple dispatching rules (SDRs) to maximize the total value of the jobs, in order to facilitate timely decision making. The experiments led us to conclude that the proposed technique is significantly superior over other approaches in terms of the total value of the jobs and the average number of sub-jobs in a job.**

Index Terms—**Lot streaming, Scheduling, Job values, Exponential deteriorating rate**

I. Introduction

In the literature, a job that consumes more time than it would have consumed if it had begun earlier is characterized as a deteriorating job. Scheduling problems with deteriorating jobs were introduced by Gupta and Gupta [14] and Browne and Yechiali [3]. Since then, the topic has received continuous attention from researchers who constructed a variety of models, in all of which the processing time of a job is a function of its starting time. Some representative deterioration functions of job processing time, such as linear, piecewise-linear and exponential functions, are common in earlier studies. Many models in the present literature are focused on the scheduling of deteriorating jobs under linear deteriorating processing time [1][2][6][8][18][20][21][23] [24][29][30][31][32]. At the same time, there are also several studies in which the job processing time deteriorates as a piecewise-linear function if its starting time is beyond a given deterioration rate [7][9][19]. Janiak and Kovalyov [16][17] considered the scheduling problems in which the processing times of the jobs are exponential functions of their starting times. Most of the above mentioned studies are based on a single-machine and flow shop setting, which is relatively unexplored in the job shop environment. Another kind of deteriorating problem is about job values deteriorating over time. For the case of job values deteriorating exponentially over time, Voutsinas and Pappis [28] developed heuristics to solve single machine scheduling problem. Raut et al. [25][26] developed several heuristics to solve the capacity constrained single machine problem with time deteriorating job values. Raut et al. [25] showed that the performance of the heuristics proposed by Voutsinas and Pappis [28] decreases with the increased size of the problem. From the above review literature, it is evident that the scheduling problem with time deteriorating job values is unexplored in the job shop environment. The present study proposes a new job shop scheduling problem, whose main characteristic consists of the critical parameter, job value, which deteriorates exponentially over time with the objective of maximizing the total value of the jobs.

This is the first time a study has attempted to extend the application of the lot streaming (LS) technique to maximize the total value of the jobs under the assumption of exponential deterioration in a job shop environment. The term LS was first introduced by Reiter [27] and later embodied by Lundrigan [22]. LS is a process of splitting jobs into smaller sub-jobs such that successive operations can be overlapped. This way, makes it possible for the job flow time to be shortened, and gaining a greater total value of the jobs. LS has been primarily applied to the flow shop problem which only allows one route for all jobs [10][12][13]. Few scholars have attempted to address the benefits of applying LS to job shop scheduling. However, some studies regarding the application of LS to job shop scheduling can be found. Chan et al. [4][5] proposed an approach using a genetic algorithm (GA) to determine the LS conditions in a job shop and in an assembly job shop environment, respectively. However, lot streaming is ignored in these studies for solving a special kind of job shop scheduling problem, where the job value is exponentially deteriorating over time.

In this study, it is assumed that a set of n jobs $J_1, J_2, \ldots, J_n$ are to be processed in a ten-machine job shop, namely M_1, M_2, …, M_{10}. The demand of J_i is given as DM_i. Each job routing is purely random, with the number of operations uniformly distributed between three and ten. Associated with each job i is its job value function $V_i(t)$, which is the value of one unit of J_i. In this study, assuming that $V_i(t)$ deteriorates

Manuscript received December 29, 2010. This work was supported by National Science Council of Taiwan under grant no. NSC 99-2221-E-020-023-MY2.

Cheng-Hsiang Liu is an Assistant Professor of Department of Industrial Management at the National Pingtung University of Science & Technology in Taiwan. He received his M.S. and Ph.D. degrees from the National Chiao Tung University (NCTU) in Taiwan, both in industrial engineering and management. His research areas include data mining, decision support systems, and operation scheduling and production control. (Tel: 886-8-7703202, ext. 7717; E-mail: chliu.uw@gmail.com).

exponentially over time,

$$V_i(t) = w_i \times e^{-\lambda_i t} \tag{1}$$

where w_i and λ_i denote the unit value at time zero and the deterioration rate of J_i, respectively. If LS is permitted, DM_i can be split into S_i sub-jobs. The size of sub-job j of job i is Q_{ij}. The processing sequence and unit processing time of the sub-jobs are the same as that of the original job. The fixed setup time (*SU*) is counted if a fixture changeover is required between jobs or sub-jobs. In that case it is best to find a sub-job combination and a processing schedule of these sub-jobs in such a way that the total value reduction of the jobs is minimized.

This study consists of two stages of development and evaluation of LS techniques. In the first stage of the proposed LS technique, a fixed-number job-splitting (FNJS) approach is employed, in which the optimal number of sub-jobs is determined by computational simulation. Then, a simple rounding procedure is applied to provide sub-jobs with an integer number of items. Given the fixed sub-job number, sequencing of the sub-jobs on the machines is determined by several simple dispatching rules (SDRs) where each sub-job is treated as an independent lot, with the objective of maximizing the total value of the jobs. Some of these dispatching rules are adaptations of existing heuristics for similar problems while others are new and specifically proposed for the problems of the type considered here. The main purpose of the first stage in this study is to examine the influence of the number of sub-jobs of a job on the performance measure (i.e., total value of the jobs) by systematically varying this parameter. In addition, this study focuses on the performance of a number of SDRs for scheduling jobs with values exponentially deteriorating over time.

In the second stage of this study, a genetic algorithm-based job splitting (GAJS) approach is developed. Using the proposed GAJS, a single solution can include the information that is needed to solve the sub-job combinations, i.e. (i) which job should be split, (ii) the number of sub-jobs, and (iii) the size of each sub-job. To facilitate timely decision making, three sub-job sizing policies are proposed to determine the size of each sub-job. Each job can either be split into a number of equal size (ES) sub-jobs or varied size (VS) sub-jobs depending on which sub-job sizing policy is chosen by the GA. Given the fixed sub-job number, sequencing of the sub-jobs on the machines is also determined by several simple dispatching rules (SDRs) where each sub-job is treated as an independent lot, with the objective of maximizing the total value of the jobs.

II. Development and Evaluation of the LS Technique

If LS is allowed, the present problem can be divided into 2 sub-problems, sub-problem 1 (SP1): determination of sub-job combinations and sub-problem 2 (SP2): job shop scheduling with all sub-jobs. Both SP1 and SP2 are complex problems. In the first stage of this study, a traditional job splitting approach and eight simple dispatching rules (SDRs) are presented to solve these two sub-problems. First, a fixed number job splitting (FNJS) approach is presented to split a job into several sub-jobs, in which the optimal fixed number of sub-jobs of a job is determined by computational simulation. Then, a simple rounding procedure is applied to provide sub-jobs with an integer number of items. For each job i, if there are q positive sub-jobs, all sub-jobs Q_{ij}, j=1, …, q-1 are rounded to the closest integer, and the last sub-job Q_{iq} is set equal to $DM_i - \sum_{j=1}^{q-1} Q_{ij}$.

This study tests a variety of dispatching rules. Some of these rules are adaptations of existing rules for similar problem while others are new and specifically proposed for the problems of the type considered here.

SRT = Priority to process a sub-job at any given machine k goes to that sub-job queuing in front of a machine having a minimum $RP_{ijk} \times Q_{ij}$, where RP_{ijk} is the remaining amount of processing time needed to complete the jth sub-job of job i on machine k

SRT.v = Priority to process a sub-job at any given machine k goes to that sub-job queuing in front of a machine having a minimum $\dfrac{RP_{ijk} \times Q_{ij}}{W_i e^{-\lambda_i t}}$

SRT.d = Priority to process a sub-job at any given machine k goes to that sub-job the queuing in front of a machine having a minimum $\dfrac{RP_{ijk} \times Q_{ij}}{W_i - W_i e^{-\lambda_i t}}$

SRT.m = Priority to process a sub-job at any given machine k goes to that sub-job queuing in front of a machine having a minimum $\dfrac{RP_{ijk} \times Q_{ij}}{W_i e^{-\lambda_i t} \times (W_i - W_i e^{-\lambda_i t})}$

LRO = Priority to process a sub-job at any given machine k goes to that sub-job queuing in front of a machine having a minimum R_{ijk} , where R_{ijk} is the set of remaining operations to complete the jth sub-job of job i on machine k

LRO.v = Priority to process a sub-job at any given machine k goes to that sub-job queuing in front of a machine having a minimum $\dfrac{R_{ijk}}{W_i e^{-\lambda_i t}}$

LRO.d = Priority to process a sub-job at any given machine k goes to that sub-job queuing in front of a machine having a minimum $\dfrac{R_{ijk}}{W_i - W_i e^{-\lambda_i t}}$

LRO.m = Priority to process a sub-job at any given machine k goes to that sub-job queuing in front of a machine having a minimum $\dfrac{R_{ijk}}{W_i e^{-\lambda_i t} \times (W_i - W_i e^{-\lambda_i t})}$

Among these proposed simple dispatching rules, SRT.v and LRO.v were derived by using additional information on the current values of sub-jobs. SRT.d and LRO.d make use of additional information on the loss in sub-job value. SRT.m and LRO.m rules simultaneous consider the current value and the loss in value when scheduling a particular sub-job at a given point in time.

The following is a description of the general job shop environment modeled in the simulation. The shop used in this

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

study is made up of ten machines. There are 30 jobs in the system. Each job requires from three to ten operations in its process routing. For each job, the sequence of operations was randomly generated with an equal probability of initially starting at any of the ten machines. Once started, a job has an equal probability of proceeding to any other nine machines —a purely random routing. The unit of processing time for each operation was taken from a uniform distribution from one to fifty time units. The demand of each job is uniformly generated over the integer set [1, 30].

In the following section, a number of experiments are carried out to examine the proposed methodology in this section. The main purpose is to address the performance of the proposed LS technique under different scenarios. A four-factor full factorial design is employed to conduct a comprehensive study of the effects of the decision factors on the performance measure. The factors to be evaluated are shown in Table 1. For each of the 72 treatments, ten replications are conducted in order to minimize the variation of the results. Thus, there are a total of 720 experiments. For each problem, the job-splitting approach together with the eight SDRs will be executed in two modes: not allowed job splitting (noJS) and FNJS. In FNJS, the number of sub-jobs (q) is varied from one to thirty.

Table 1. Decision factor setting

A. Factor	Level
Setup time	30, 60
Deterioration rate	Low, $\lambda_i \in$ Uniform [0, 0.2]
	Medium, $\lambda_i \in$ Uniform [0, 0.4]
	High, $\lambda_i \in$ Uniform [0, 0.6]
Job splitting approach	noJS, FNJS
Simple dispatching rule	SRT, SRT.v, SRT.d, SRT.m, LRO, LRO.v, LRO.d, LRO.m

The performances of the eight SDRs and two job splitting approaches (noLS and FNLS) under study are evaluated with respect to total value of the jobs. Our key observations are summarized below.

(1) It is evident that FNJS substantially increases the total value of the jobs over noJS for all tested conditions.
(2) For a low deterioration rate, it is evident that LRO.m, in conjunction with FNJS, outperforms the rest. They are combined to obtain the maximum total value of jobs. When the level of deterioration rate increases, SRT.m in conjunction with FNJS clearly becomes the dominant approach.
(3) On average, the SRT-based rules perform better than the LRO-based rules.
(4) The SDRs (LRO.m and SRT.m) that simultaneously consider the current value and the loss in value surpass the SDRs that only consider either the current value (LRO.v and SRT.v) or the loss in value (LRO.d and SRT.d).
(5) The relative performances of LRO.v, LRO.d, SRT.v and SRT.d depend on the deterioration rate. With a low deterioration rate, LRO.v and SRT.v are superior to their counterparts (LRO.d and SRT.d), respectively. When the deterioration rate increases, the tendency is for LRO.d and SRT.v to outperform their counterparts, LRO.v and SRT.v, respectively.
(6) Increasing the setup time may diminish the effects of the LS techniques.

Furthermore, from the study of the average number of sub-jobs of a job required to maximize the total value of the jobs by FNJS, two main observations are made:

(1) The number of sub-jobs in a job increases with the increase in the deterioration rate, except for the FNJS in conjunction with SRT-based rules.
(2) The FNJS approach tends to split jobs into fewer sub-jobs if the setup time increases.

Based on the above experimental results it can be concluded that our proposed LS technique consisting of FNJS and SDRs which make use of the current value and the loss in value information can significantly increase the total value of the jobs over noJS in conjunction with traditional dispatching rules (i.e., LRO and SRT) under different deterioration rates and setup times. Furthermore, the number of sub-jobs of a job tends to increase when the level of deterioration rate becomes greater.

III. GENETIC ALGORITHM BASED JOB SPLITTING APPROACH (GAJS)

This section, proposes an effective approach to solve the sub-job combinations, i.e. (i) the number of sub-jobs of a job and (ii) the size of each sub-job, using the genetic algorithm (GA) and three proposed sub-job sizing policies. The original idea of the GA was developed independently by both Holland [15] and DeJong [11]. Let's consider the case where a GA is to be used to solve a lot streaming problem with job values exponentially deteriorating over time. First, the chromosome structure of the GA must be defined. Consider for example a problem consisting of five jobs. Figure 1 describes the proposed chromosome structure in a two-dimension (5×2) matrix.

$$\begin{bmatrix} 3 & 1 \\ 1 & 2 \\ 5 & 1 \\ 4 & 3 \\ 2 & 1 \end{bmatrix}$$

Figure 1. The chromosome structure of the proposed GA

where if X_{i1}=1, it means that no splitting is applied to the job. Otherwise, if X_{i1}>1, it means that the job is split into X_{i1} sub-jobs. We then enlarge the domain for each gene X_{i2} to the values [1, 2, 3] and define its meaning as follows:

$$value\ of\ X_{i2} = \begin{cases} 1, \textit{Equal Size Policy} \\ 2, \textit{Gradient Varied Size Policy} \\ 3, \textit{Three Levels Varied Size Policy} \end{cases}$$

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Equal-Size Policy

$If \left(\frac{DM_i}{S_i}\right) \text{ is not an Integer Then}$

$$Q_{ij} = Floor\left(\frac{DM_i}{S_i}\right),\ j = 1, ..., S_i - 1$$

$$Q_{ij} = DM_i - \sum_{j=1}^{S_i - 1} Q_{ij},\ j = S_i$$

Else

$$Q_{ij} = \frac{DM_i}{S_i}$$

End If

Gradient Varied-Size Policy

$$Q_{ij} = Floor\left(\frac{\frac{j}{S_i}}{\frac{1}{2}\times(1+S_i)} \times DM_i\right),\ j = 1, ..., S_i - 1$$

$$Q_{ij} = DM_i - \sum_{j=1}^{S_i - 1} Q_{ij},\ j = S_i$$

Three Levels Varied-Size Policy

$If \left(\frac{DM_i}{3}\right) \text{ is not an Integer Then}$

$$Y = 2S_i + 1$$

Else

$$Y = 2S_i$$

End If

$$Q_{ij} = \left(\frac{1}{Y}\right) \times DM_i,\ j < S_i \text{ and } \frac{j}{S_i} \le 0.33$$

$$Q_{ij} = \left(\frac{2}{Y}\right) \times DM_i,\ j < S_i \text{ and } 0.33 < \frac{j}{S_i} \le 0.66$$

$$Q_{ij} = \left(\frac{3}{Y}\right) \times DM_i,\ j < S_i \text{ and } \frac{j}{S_i} > 0.66$$

$$Q_{ij} = DM_i - \sum_{j=1}^{S_i - 1} Q_{ij},\ j = S_i$$

Based on the above description, the proposed genetic algorithm-based job splitting (GAJS) approach not only allows splitting a job into a number of equal-sized sub-jobs, but also into a number of varied-sized sub-jobs.

The following operations describe one generation of a GA. The fitness of each chromosome is assessed by computing its objective function, the total value of the jobs. Once the fitness value of each chromosome has been assessed, the Roulette Wheel Selection is implemented to select chromosomes for the crossover operation. First two parent chromosomes are selected. Some feasible subset of genes is then swapped between two parents, producing two new offspring chromosomes. After the crossover operation, mutation takes place subject to the probability of introducing new genes within the selected chromosome, to propagate offspring with more diverse characteristics. The fitness value of the offspring is also assessed. Elitism, which refers to the best φ% of the population being transferred from the previous generation to the current generation is employed. Figure 2 depicts the crossover operation used with this chromosome structure for a six order example. In Figure 2a, the third row has been selected randomly and has been exchanged between Parent 1 and Parent 2. Figure 2b shows a mutation operation for the above described example. It also shows how the fourth row of chromosomes is mutated. The procedure is repeated until the terminating criteria are met. Then, after splitting jobs into sub-jobs, we need to solve the job shop scheduling problem, which is also NP-hard. To solve this problem, our proposed SDRs which are simple, time-saving and effective are applied again.

$$\begin{array}{ccc} Parent\ 1 & Offspring\ 1 & Offspring\ 1 \\ \begin{bmatrix} 3 & 1 \\ 5 & 2 \\ 1 & 3 \\ 2 & 1 \\ 4 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 1 \\ 1 & 2 \\ 5 & 1 \\ 2 & 1 \\ 4 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 1 \\ 1 & 2 \\ 5 & 1 \\ 2 & 1 \\ 4 & 2 \end{bmatrix} \\ & (a) & (b) \\ & \Rightarrow & \Rightarrow \\ Parent\ 2 & Offspring\ 2 & Offspring\ 2 \\ \begin{bmatrix} 4 & 3 \\ 1 & 3 \\ 5 & 1 \\ 2 & 1 \\ 3 & 2 \end{bmatrix} & \begin{bmatrix} 4 & 3 \\ 5 & 3 \\ 1 & 3 \\ 2 & 1 \\ 3 & 2 \end{bmatrix} & \begin{bmatrix} 4 & 3 \\ 5 & 3 \\ 1 & 3 \\ 3 & 2 \\ 2 & 2 \end{bmatrix} \end{array}$$

Figure 2. (a) crossover and (b) mutation operation of the proposed GA.

IV. COMPARISON OF GAJS TO FNJS

To demonstrate the effectiveness of the proposed GAJS approach for job shop scheduling with job values exponentially deteriorating over time, an extensive set of problems with different characteristics were generated for the total value of the jobs. A three-factor full factorial design was employed to conduct a comprehensive study of the effects of the decision factors on the performance measure. The factors to be evaluated were deterioration rate, setup time, and job splitting approach. For each treatment, ten replications were conducted in order to minimize the variation of the results. Under each factor combination of deterioration rate and setup time, the best SDR was chosen to solve the JSP. Based on the experimental results in Section II, the LRO.m rule performed best for the total value of the jobs under a low deterioration rate, but when the deterioration rate increased, the SRT.m rule became dominant. Therefore, LRO.m and SRT.m rules were included in the second stage of experiments. The decision as to which one to use is based on the level of deterioration.

After some preliminary tests, the GA parameters were set to a population size (PS) = 100, crossover rate = 0.80, and mutation rate = 0.05. The procedure of GAJS approach continued until the terminating criterion was reached, which specifies to terminate if after 30 consecutive generations there is no further performance improvement.

Tables 2-3 present the total value of the jobs yielded by the GAJS and FNJS approaches at different setup time values. Based on these tables, the performance of the GAJS approach was very good with respect to maximizing the total value of the jobs. The improvement of GAJS over FNJS increased when the deterioration rate increased. In addition, the effect of GAJS was greater than the effect of FNJS when the setup time increased from 30 to 60. In this study we also recorded the number of sub-jobs of a job under each factor

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

combination. From the study of the average number of sub-jobs of a job using GAJS and FNJS, it was evident that the GAJS approach tends to split jobs into a lesser number of sub-jobs than the FNJS approach does.

Table 2. The simulation results for GAJS and FNJS at setup time=30

Deterioration Rate	Dispatching Rule	Job Splitting Approach		% Diff.
		GAJS	FNJS	
Low	LRO.m	639223.61	627442.97	1.88%
Medium	SRT.m	520657.46	500548.76	4.02%
High	SRT.m	439337.73	419189.30	4.81%

Table 3. The simulation results for GAJS and FNJS at setup time=60

Deteriorating Rate	Dispatching Rule	Job Splitting Approach		% Diff.
		GAJS	FNJS	
Low	LRO.m	620856.75	604244.89	2.75%
Medium	SRT.m	494440.86	471534.76	4.86%
High	SRT.m	414307.09	392080.28	5.67%

V. Conclusions

This study is the first to extend lot streaming to the job shop scheduling problem with job value exponentially deteriorating over time. To facilitate timely decision making, the proposed GA determines the sub-job combinations (sub-problems 1 and 2) and the SDRs are applied to solve the job shop scheduling problem with all sub-jobs. Within the proposed genetic algorithm, named the GAJS, a single chromosome can hold the information that we need to solve (i) which job should be split and the number of sub-jobs of a job and (ii) given a fixed number of sub-jobs, which sub-job sizing policy should be used to determine the size of each sub-job. The GAJS approach not only allows splitting a job into a number of equal-sized sub-jobs, but also a number of various-sized sub-jobs. In addition, using SDRs, the job shop scheduling problem with all sub-jobs can be solved quickly.

The results of the experiment suggest that dividing each job into several sub-jobs allows for the improvement of the total value of the jobs. The SDRs (LRO.m and SRT.m) that simultaneously consider the current value and the loss in value are superior to the SDRs that only consider either the current value (LRO.v and SRT.v) or the loss in value (LRO.d and SRT.d). Under the FNJS approach, the number of sub-jobs of a job increases as the level of deterioration rate increases, except for the cases where the FNJS approach is used together with SRT-based rules. In addition, FNJS tends to split jobs into a lesser number of sub-jobs if the setup time increases. With respect to the maximization of the total value of the jobs the GAJS approach with the LRO.m rule emerges as being the best in the case of a low deterioration rate, whereas for medium and high deterioration rates, the GAJS in conjunction with the SRT.m rule provides the best performance. The GAJS splits a job into a smaller number of sub-jobs than the FNJS approach, which benefits the production manager. In summary, the findings of this study indicate that GAJS is superior over FNJS in terms of the number of sub-jobs and the total value of the jobs. Therefore, it would be advisable for a production manager to adopt the GAJS method rather than the FNJS method for splitting a job into sub-jobs.

References

[1] A. Bachman and A. Janiak, “Minimizing maximum lateness under linear deterioration,” *European Journal of Operational Research*, vol. 126, no. 3, pp. 557-566, 2000.

[2] A. Bachman, A. Janiak, and M. Y. Kovalyov, “Minimizing the total weighted completion time of deteriorating jobs,” *Information Processing Letters*, vol. 81, no. 2, pp. 81-84, 2002.

[3] S. Browne and U. Yechiali, “Scheduling deteriorating jobs on a single processor,” *Operations Research*, vol. 38, pp. 495-498, 1990.

[4] F. T. S. Chan, T. C. Wong, and L. Y. Chan, “Lot streaming for product assembly in job shop environment,” *Robotics and Computer-Integrated Manufacturing*, vol. 24, pp. 321-331, 2008.

[5] F. T. S. Chan, T. C. Wong, and L. Y. Chan, “The application of genetic algorithms to lot streaming in a job-shop scheduling problem,” *International Journal of Production Research*, vol. 47, no. 12, pp. 3387-3412, 2009.

[6] T. C. E. Cheng and Q. Ding, “The complexity of single machine scheduling with release times,” *Information Processing Letters*, vol. 65, no. 2, pp. 75-79, 1998.

[7] T. C. E. Cheng and Q. Ding, “Single machine scheduling with step-deteriorating processing times,” *European Journal of Operational Research*, vol. 134, pp. 623-630, 2001.

[8] T. C. E. Cheng and Q. Ding, “Scheduling start time dependent tasks with deadlines and identical initial processing times on a single machine,” *Computers & Operations Research*, vol. 30, no. 1, pp. 51-62, 2003.

[9] T. C. E. Cheng, Q. Ding, M. Y. Kovalyov, A. Bachman, and A. Janiak, “Scheduling jobs with piecewise linear decreasing processing times,” *Naval Research Logistics*, vol. 50, no. 6, pp. 531-554, 2003.

[10] F. M. Defersha and M. Chen, “A hybrid genetic algorithm for flowshop lot streaming with setups and variable sublots,” *International Journal of Production Research*, vol. 48, no. 6, pp. 1705-1726, 2010.

[11] K. DeJong, “An analysis of the behavior of a class of genetic adaptive systems,” Ph.D. dissertation, University of Michigan, Ann Arbor, 1975.

[12] M. Feldmann and D. Biskup, “Lot streaming in a multiple product permutation flow shop with intermingling,” *International Journal of Production Research*, vol. 46, no. 1, pp. 197-216, 2006.

[13] C. A. Glass and Y. T. Herer, “On the equivalence of small batch assembly line balancing and lot streaming in a flow shop,” *International Journal of Production Research*, vol. 44, no. 21, pp. 4587-4606, 2006.

[14] J. N. D. Gupta and S. K. Gupta, “Single facility scheduling with nonlinear processing times,” *Computers & Industrial Engineering*, vol. 14, pp. 387-393, 1988.

[15] J. Holland, “Adaptation in natural and artificial systems,” MI: The University of Michigan Press, Ann Arbor, 1975.

[16] A. Janiak and M. Y. Kovalyov, “Scheduling in a contaminated area: A model and polynomial algorithms,”. *European Journal of Operational Research*, vol. 173, no. 1, pp. 125-132, 2006a.

[17] A. Janiak and M. Y. Kovalyov, “Job sequencing with exponential functions of processing times,” *Informatica*, vol. 17, no. 1, pp. 13-24, 2006b.

[18] A. Kononov and S. Gawiejnowicz, “NP-hard cases in scheduling deteriorating jobs on dedicated machines,” *Journal of the Operational Research Society*, vol. 52, no. 6, pp. 708-818, 2001.

[19] A. S. Kunnathur and S. K. Gupta, “Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem,” *European Journal of Operational Research*, vol. 47, no. 1, pp. 56-64, 1994.

[20] W.-C. Lee, Y.-R. Shiau, S.-K. Chen, and C.-C. Wu, “A two-machine flowshop scheduling problem with deteriorating jobs and blocking,” *International Journal of Production Economics*, vol. 124, no. 1, pp. 188-197, 2010.

[21] W.-C. Lee, C.-C. Wu, C.-C. Wen, and Y.-H. Chung, “A two-machine flowshop scheduling problem with deteriorating jobs,” *Computers & Industrial Engineering*, vol. 54, pp. 737-749, 2008.

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[22] R. Lundrigan, “What is this called OPT?,” *Production Inventory Management Journal*, vol. 27, pp. 2-11, 1986.
[23] G. Mosheiov, “Λ-shaped policies to schedule deteriorating jobs,” *Journal of the Operational Research Society*, vol. 47, pp. 1184-1191, 1996.
[24] C. T. Ng, T. C. E. Cheng, A. Bachman, and A. Janiak, “Three scheduling problems with deteriorating jobs to minimize the total completion time,” *Information Processing Letters*, vol. 81, no. 6, pp. 327-333, 2002.
[25] S. Raut, J. N. D. Gupta, and S. Swami, “Single machine scheduling with time deteriorating job values,” *Journal of Operational Research Society*, vol. 59, pp. 105-118, 2008.
[26] S. Raut, , S. Swami, and J. N. D. Gupta, “Scheduling a capacitated single machine with time deteriorating job values,” *International Journal of Production Economics*, vol. 114, pp. 769-780, 2008.
[27] S. Reiter, “A system for managing job shop production,” *Journal of Business*, vol. 34, pp. 371-393, 1966.
[28] T. G. Voutsinas and C. P. Pappis, “Scheduling jobs with values exponentially deteriorating over time,” *International Journal of Production Economics*, vol. 79, pp. 163-169, 2002.
[29] J. B. Wang and Z. Q. Xia, “Flow shop scheduling problems with deteriorating jobs under dominating machines,” *Journal of the Operational Research Society*, vol. 57, pp. 220-226, 2006.
[30] J. B. Wang, A. X. Guo, F. Shan, B. Jiang, and L. Y. Wang, “Single machine group scheduling under decreasing linear deterioration,” *Journal of Applied Mathematics and Computing*, vol. 24, pp. 283-293, 2007.
[31] J. B. Wang, C. T. D. Ng, T. C. E. Cheng, and L. L. Liu, “Minimizing total completion time in a two-machine flow shop with deteriorating jobs,” *Applied Mathematics and Computation*, vol. 180, pp. 185-193, 2006.
[32] C. C. Wu and W. C. Lee, “Two-machine flowshop scheduling to minimize mean flow time under linear deterioration,” *International Journal of Production Economics*, vol. 103, pp. 572-584, 2006.

ISBN: 978-988-19251-2-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)