A Simulated Annealing Algorithm for Single Machine Scheduling Problem with Release Dates, Learning and Deteriorating Effects

Sergio Fichera, Fulvio Cappadonna, Antonio Costa, Alberto Fichera

Abstract— **In this paper the single machine total weighted completion time scheduling problem is studied. The jobs have non zero release time and processing time increases during the production due to the effect of deterioration on the machine. The processing time can decrease due the learning capacity of the worker. A simulated annealing heuristic algorithm is proposed to solve the scheduling problem, and their efficiency is evaluated on a benchmark.**

Index Terms— **Single machine, Release dates, Heuristic, Simulated annealing, deteriorating, workforce, learning**

I. Introduction

To be competitive in the worldwide market of goods, companies should produce small lots of different products at a convenient cost level and different quality standards. Consequently, manufacturing systems should be flexible and reconfigurable in a short time. In this ever-changing environment, the workforce plays a strategic role: thus, their learning ability should be considered. Moreover, the deterioration effect on the machine must be considered to cope with more realistic conditions. Recent literature about single machine scheduling proposes models embedding workforce and how workers can interact with the scheduling strategy. The aim of the proposed paper is to investigate the scheduling of a single machine managed by a workforce that has learning abilities. The model considers the scheduling of weighted jobs on a single machine, each having nonzero release date. The processing time of the jobs increases during production due the deterioration of the machine. The workforce ability influences the working time of each scheduled job. The objective of the scheduling is the total weighted completion time.

Currently, to the best of our knowledge no heuristic optimization procedure has been proposed in literature to solve this kind of problem. The aim of this paper is to propose an heuristic algorithm to find efficient solutions to the scheduling problem of jobs.

The rest of the paper is organized as follows: in Section 2, we will present a literature review concerning the studied problem. In order to solve the scheduling problem the problem formulation and the heuristic algorithm are presented in Section 3.The results of computational experiments are given in Section 4. The conclusions of the research are summarized in section 5.

S. Fichera is with the University of Catania, viale A. Doria 6, 95125 Catania, Italy (e-mail: sfichera@dii.unict.it).

F. Cappadonna is with the University of Catania, viale A. Doria 6, 95125 Catania, Italy (e-mail: fcappadonna@dieei.unict.it).

A. Costa is with the University of Catania, viale A. Doria 6, 95125 Catania, Italy (e-mail: costa@dii.unict.it).

A. Fichera is with the University of Catania, viale A. Doria 6, 95125 Catania, Italy (e-mail: afichera@dii.unict.it).

II. Literature review

The first studies about learning effect had been developed by Wright [1]. Biskup [2] proposed that the production time of a job under learning effect decreases depending on the order the job is worked in. He introduced a learning effect model in which the processing time of job J_j when it is scheduled in the r_{th} position in a processing sequence is defined as

$$p^A_{j[r]} = p_j r^a \qquad (1)$$

where p_j is the normal processing time of job J_j and $a = \log_2 LR < 0$ is the learning index, which is a function of the learning rate LR<1. The processing time needed decreases by the number of repetitions, meaning that learning is primarily based on the repetition of a task, such as machine setup.

In alternative to the above position based learning model Kuo and Yang [3] proposed a sum-of-processing-time based learning model which is time depended. Starting from these two fundamental models, some authors proposed other learning models that are a modification of the above models or their combination Cheng, Wu, and Lee [4], Yin, Xu, Sun, and Li [5]. Biskup [6], reviews extensively the literature on scheduling problems that considers the two types of learning effects. In addition to learning effect many authors introduced in the scheduling problem the effect of jobs deterioration, i.e., jobs whose processing times are increasing functions of their starting times due the deterioration of the machines or the delay of maintenance or cleaning Mosheiov [7]

Tamer Eren [8] considers in a single machine scheduling problem the learning effect influencing the job set-up time and the deterioration influencing the job processing time separately. Both are modeled as position time dependent. In single machine scheduling problem in order to consider a more general condition the assumption of unequal release of the jobs is due. Lee et al. [9] studied a single-machine position-based learning scheduling problem with release times where the objective is to minimize the sum of makespan and total completion time. Based on the aforementioned literature review a single machine

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

scheduling problem with unequal release date of jobs, worker with learning ability and the machine with deteriorating effect where the objective is the total weighted completion time is studied in this paper.

III. PROBLEM FORMULATION

The investigated problem considers a single machine scheduling of weighted jobs with different release dates. The learning effect is modeled as proposed Eren [10] Accordingly to this model the processing time of job *j* varies with learning and is equal to $PTj_{jk}= PT_j*k^a$ where PT_j is the needed processing time for the *k*-th worked unit and $a = log_2LR \leq 0$ is the *learning index*, which is a function of the learning rate *LR*. The deterioration effect is modeled in the same manner as the learning index, but the deterioration index is $b\geq 0$.

The goal of the problem is to find the best sequence of jobs in order to minimize the Total Weighted Completion Time.

IV. SIMULATED ANNEALING HEURISTIC

The simulated annealing algorithm works on a sequence Seq, representing a feasible sequencing of jobs. It starts with an initial seed sequence and then it evolves through a selected number of iterations with the aim of minimizing the value of the objective function OBJ. Each iteration represents a temperature stage which the algorithm goes through during its cooling process: thus, the first iteration corresponds to the upper temperature, or initial temperature T_0. The simulated annealing investigates throughout the space of feasible solutions by applying a proper shift insertion neighbourhood search scheme on the actual sequence Seq. The shift insertion selects two generic positions and r, (s>r), within the sequence; then it extracts the element in the s*th* position and moves the elements between s+1 and r positions to the left; finally, it positions the extracted element in position r.

A cooling schedule based on the variation of the objective function and on the actual temperature stage has been chosen. The current sequence is perturbed according to the neighborhood search scheme proposed above. If the new evaluated vector Seq' has an objective function whose value is lower than the one corresponding to original vector Seq, it is accepted and becomes the new reference sequence. On the other hand, if the vector Seq' has a worse value of the objective function it is accepted only if equation (9) is verified:

$$rand(x) < exp\left(-\frac{(OBJ(Seq')-OBJ(Seq))}{T_i}\right) \quad (9)$$

where x is a random variable having uniform distribution U[0; 1].

In this way, the probability of accepting a worst sequence Seq', that is a sequence corresponding to a larger value of the total weighted completion time, is not constant during the algorithm evolution. It reduces in accordance with the value of the actual temperature T_i. At the low temperatures, when good solutions have been achieved and the probability of accepting a worse sequence is significantly reduced, the algorithm risks remaining blocked at a temperature level for a long computing time. To overcome this problem a counter N_{max} limiting the number of sequences investigated at each temperature stage T_i has been introduced; when N_{max} is reached, T_i is reduced even if the equation (9) is not satisfied.

The algorithm stops when the temperature stage Ti achieves the final value T_f or when a fixed number 'end count' of sequences has been investigated. At the end of the simulated annealing evolution, the sequence corresponding to the minimum value of the total weighted completion time performance measure is assumed as the best schedule of the jobs to be worked.

Before starting the experimental design a preliminary tuning of the simulated annealing was performed to determine the optimal selection of the cooling parameters: the initial and final temperatures T_0 and T_f; the constant α (0≤α<1), of the cooling geometric ratio Ti+1 =α * Ti, (i=0,1,2..); the number N_{max} of investigated sequences at each temperature stage and the 'end count', that is the condition which finishes the algorithm execution. After a set of statistical analyses performed on different sized problems and considering several algorithm runs starting from different seed values, the following optimal algorithm configuration was determined. This works well for all the considered scenarios both in terms of convergence of solution and control of the computational times: T_i= 500, T_f = 2, α = 0.99, N_{max} = 250 and 'end count' = 20,000 investigated sequences. The following step-by-step procedure describes the logic of the heuristic procedure based on the simulated annealing:

Step 1: Initialize: $T_i=T_0$, T_f, α, N_{max} and N=1.

Step 2: Randomly create a feasible seed sequence and assign it to ***Seq*** vector.

Step 3: Evaluate the *OBJ*(***Seq***) value for ***Seq*** by running a simulation and assign it to variable ***BEST***.

Step 4: Generate a perturbed sequence ***Seq'*** with one the neighborhoods operators and set *N=N+1*.

Step 5: Evaluate the *OBJ*(***Seq'***) value for ***Seq'*** by running a simulation.

Step 6: If *OBJ*(***Seq'***)<***BEST*** then let ***Seq***=***Seq'***, ***BEST***=*OBJ*(***Seq'***). Set N=N_{max} and go to Step 9.

Step 7: Generate a random number x.

Step 8: If rand(x)<exp(-(*OBJ*(***Seq'***)-OBJ(***Seq***))/T), then let ***Seq***=***Seq'***. Set $N=N_{max}$

Step 9: If $N<N_{max}$ then go to step 4

else

let $T_{i+1}= \alpha\ T_i$ and *N*=0.

Step 10: If $T>T_f$, then go to Step 4.

Step 11: STOP. Best value of *OBJ* stored in ***BEST***.

V. COMPUTATIONAL RESULTS

In order to evaluate the efficiency of the proposed algorithm a benchmark of problems has been investigated. The processing times are drawn at random from the discrete uniform distribution *U*[1,100]. The weights are drawn at random from the discrete uniform distribution *U*[1,10]. The release dates are uniformly distributed between [0.5; 0.5nR]

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

where *R*, a control variable, is a value out from a specific set: 0.2, 0.4, 0.6, 0.8, 1.0, 1.25, 1.5, 1.75, 2.0, 3.0. The learning index is a random variable *U*[-0.4,0] and the deterioration is randomly distributed as *U*[0,0.4.]. A class of problem is defined by a number of job and a specific value of R, for each class twenty instances are generated. The benchmark is featured by 10 classes of problems, (10, 20, 30, 40, 50, 60, 70, 80, 90 and 100 jobs), for each class ten value of the control variable R, for each combination job and control variable R there are 20 instances. In total the benchmark included 2000 instances for fixed values of learning and deterioration indexes.

The above classes of examples have been solved through the simulated annealing algorithm. The value of learning index is equal to -0.35 and the deterioration index is 0.01.

The results obtained performing the classes from 10 to 100 jobs are compared with a local search heuristic to evaluate the performance of the simulated annealing algorithm (Tab. 1). The heuristic algorithms are coded in VBA language.

Table 1.
Results of 10-100 jobs scheduling problems

jobs	Local search	Number of best performance	Simulated annealing	Number of best performance
10	65461	42	64881	90
20	89961	30	89557	85
30	194146	47	193312	112
40	341359	55	339957	125
50	468775	66	466987	124
60	598069	87	596090	113
70	908152	77	904512	122
80	1366439	26	1354500	174
90	1714526	35	1702907	165
100	2249048	5	2128703	195

The results show the effectiveness of the Simulated Annealing in comparison with the Local search heuristic.

VI. Conclusion

In this paper the scheduling of a single machine managed by workforce having learning abilities is considered. Moreover, the model considers the scheduling of weighted jobs with nonzero release times and increasing processing time due to the deterioration of the machine with the ongoing production. The objective of the scheduling is the minimization of the total weighted completion time. A simulated annealing algorithm is considered to find efficient solutions. The obtained results show the efficiency of the simulated annealing algorithm to solve problems small and great problems.

References

[1] T.P. Wright, "Factors affecting the cost of airplanes", *J. Aeronaut. Sci. VOL:*3 pp. 122–128, 1936.
[2] A. Biskup, "Single-machine scheduling with learning considerations", *Eur. J. Oper. Res*. Vol. 115 pp. 173–178, 1999.
[3] W.H. Kuo, D.L. Yang. "Minimizing the total completion time in a single machine scheduling problem with a time-dependent learning effect", *European J Oper Res*; Vol. 174, pp. 1184–90, 2006.
[4] T. C. E. Cheng, , W. C. Wu, C.C. Lee, "Some scheduling problems with deteriorating jobs and learning effects", *Computers & Industrial Engineering*, Vol. 54, pp. 972–982, 2008
[5] Yin, Xu, Sun, and Li, "Single-machine scheduling problems with deteriorating jobs and learning effects" *Computers & Industrial Engineering* Vol. 57 pp. 843–846, 2009.
[6] A. Biskup,). "A state-of-the-art review on scheduling with learning effect", *European Journal of Operational Research*, Vol. 188, pp. 315–329, 2008.
[7] G. Mosheiov, "Scheduling jobs under simple linear deterioration", *Computers and Operations Research* Vol. 21, pp. 653–9, 1994.
[8] Eren T. Human and Machine Effects in a Just-In-Time Scheduling Problem Human Factors and Ergonomics in Manufacturing, Vol. 19 (4) 294–299 (2009)
[9] WC Lee, CC Wu, PH Hsu, "A single-machine learning effect scheduling problem with release times" *Omega: The International Journal of Management Science*, Vol. 38, pp. 3–11. 2010
[10] T. Eren, "Minimizing the total weighted completion time on a single machine scheduling with release dates and a learning effect" *Applied Mathematics and Computation*, Vol. 208, pp. 355–8, 2009.

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)