MemeticAlgorithmand Genetic Algorithm for the Single Machine Scheduling Problem with Linear Earliness and Quadratic Tardiness Costs

Hussam A.A. Mohammed
Department of Mathematics,
College of Education,
University of Kerbala,
hussammath@yahoo.com

Adil S. Hassan
Department of Chemistry,
College of Education,
University of Kerbala,
kar.dail@yahoo.com

Mohammed H. Saloomi
Department of Mathematics,
College of Education,
University of Kerbala,
mohammedsaloomi@yahoo.com

Ghassan A. Khtan
Department of Mathematics,
College of Education,
University of Kerbala,
ghassanmath@yahoo.com

Abstract

The Single Machine Scheduling (SMS) problem with Multiple Objective Function (MOF) is one of the most representative problems in the scheduling area. In this paper, we consider the SMS problem with linear earliness and quadratic tardiness costs, and no machine idle time. The chosen method is based on memetic algorithm and genetic algorithm.

For this purpose, Genetic Algorithms (GA) are a population-based Meta heuristics. They have been successfully applied to many optimization problems. A Memetic Algorithm (MA) is an extension of the traditional genetic algorithm. And we introduce two types of crossover. The methods were tested and various experimental results show that MA performs better than the GA for big jobs but GA was better with small jobs.

المستخلص

مسألة جدولة الماكنةِ الوحيدةِ (SMS) مع دالة متعددة الأهداف واحدة من أكثر المسائل النموذجية في نطاقِ الجدولة. في هذه البحث، نَعتبرُ مسألة جدولة الماكنةِ الوحيدةِ بالتبكيرِ الخطيِّ وكلفِ التأخرِ من الدرجة الثانيةِ، ولاوجودلفترةَ توقف على ماكنةِ. إنّ الطريقةَ المُختَارةَ مستندة على خوارزميةِ ميميتيك وخوارزمية وراثية.

لهذا الغرضِ، خوارزميات وراثية (GA) مع قاعدة مجتمع متعدد التنقيب. نحن قُدّمنا إلى العديد مِنْ مسائل تحقيقِ الأمثلية بنجاح. خوارزمية ميميتيك (MA) امتداد لخوارزميةِ الوراثيةِ التقليديةِ. كذلك نُقدّمُإثنانمِنْأنواعِالانتقالِ. الطرقكَانتْتَنائِجَتجريبيةمُجرّبةَومُخْتَلِفةَ تبين بأنّMAيُؤدّيأفضلمِنْGAللأعمال الكبيرةلكنGAكَانتْأفضلَبالأعمال الصغيرةِ.

Keywords: scheduling, single machine, linear earliness, quadratic tardiness,memetic algorithm and genetic algorithm.

1.Introduction

In this paper, we consider a single machine scheduling problem with linearearliness and quadratic tardiness costs, and no machine idle time $1 \parallel \sum_{j=1}^{n}(E_j + T_j^2)$. Single machine scheduling environments actually occur in many practicaloperations (for a recent example in the chemical industry, see Wagner et al. (2002)). Moreover, the performance of many production systems is frequentlydetermined by the quality of the schedules for a single bottleneck machine.Single processor models are then most useful in practice for scheduling such amachine. Also, the analysis of single machine problems provides results andinsights that can often be applied to more complex scheduling environments.Indeed, multiple processor environments can often be relaxed to a singlemachine problem, or a sequence of such problems. Furthermore, the solutionprocedures for some complex systems (e.g., job shops) often require solvingsubproblems with a single processor.

Scheduling models with both earliness and tardiness costs are compatiblewith the just-in-time (JIT) production philosophy. The JIT approach focuseson producing goods only when they are needed, and therefore considers thatboth earliness and tardiness should be discouraged. Earliness/tardiness modelsare also compatible with the recent adoption of supply chain managementby many organizations. This approach seeks to improve the efficiency of thesupply chain, and to provide a better service to the end user, by integratingthe flow of materials from suppliers to customers. The adoption of supplychain management has caused organizations to view early deliveries, in additionto tardy deliveries, as undesirable.

Linear earliness and quadratic tardiness costs are considered in this paper.On the one hand, early completions of jobs result in unnecessary inventory.The costs of maintaining and managing this inventory tend to be proportionalto the quantity held in stock, and therefore a linear penalty is used forearly jobs. On the other hand, late deliveries can result in lost sales and lossof goodwill, as well as disruptions in stages further down the supply chain. Inthis paper, a quadratic penalty is considered for the tardy jobs. A quadratictardiness penalty is appropriate in practice. Indeed, the tardiness is an importantattribute of service quality. Also, a customer's dissatisfaction tendsto increase quadratically with the tardiness, as proposed in the loss function of Taguchi (1986). Moreover, a quadratic tardiness penalty can in some situations be preferable to the more usual linear tardiness or maximum tardiness functions, as discussed in Sun et al. (1999).

We assume that machine idle time is not allowed. This assumption isappropriate for many production settings. Indeed, when the capacity of themachine is limited when compared with the demand, the machine must bekept running in order to meet the customers' orders. Also, the assumptionof no idle time is justified when the machine has high operating costs, andwhen starting a new production run involves large setup costs or times. Somespecific examples of production settings where the no idle time assumption is appropriate have been given by Korman (1994) and Landis (1993).

This problem has been previously considered by Valente (to appear, 2006). Valente (to appear) proposed a lower bounding procedure based on arelaxation of the job completion times, as well as a branch-and-bound procedure.In Valente (2006), several dispatching heuristics are presented, andtheir performance is analysed on a wide range of instance types. The correspondingproblem with inserted idle time has been considered by Schaller (2004). He presented a timetabling procedure to optimally insert idle time ina given sequence, as well as a branch-and-bound procedure and simple andefficient heuristics.

The single machine problem with linear earliness and tardiness penalties$\sum_{j=1}^{n}\left(E_j + T_j\right)$ has also been previously considered by Garey et al. (1988),Kim and Yano (1994) and Schaller (2007). Garey et al. (1988) showed thatthe problem is NP-hard, and proposed a timetabling procedure. Severalproperties of optimal solutions were presented by Kim and Yano (1994), andused to develop optimal and heuristic algorithms. Schaller (2007) developsa new lower bound and a new dominance condition, and also shows how tostrengthen the lower bounds proposed by Kim and Yano (1994).Mohammed (to appear, 2012) add due-dates for the problem and introduced six types for crossovers three of them new.

The minimization of the quadratic lateness $\sum_{j=1}^{n} L_j^2$, where the latenessof s_j is defined as $L_j = C_j - d_j$, has also been previously considered. Guptaand Sen (1983) proposed both a branch-and-bound algorithm and a heuristic rule for the problem with no idle time. Su and Chang (1998) and Schaller (2002) considered inserted idle time, and proposed timetabling proceduresand heuristic algorithms. Sen et al. (1995) presented a branch-and-boundprocedure for the weighted problem$\sum_{j=1}^{n} w_j L_j^2$ where idle time is allowedonly prior to the start of the first job. Baker and Scudder (1990) andHoogeveen (2005) provide excellent surveys of scheduling problems with earlinessand tardiness penalties. Kanet and Sridharan (2000) give a review ofscheduling models with inserted idle time that complements our focus on aproblem with no machine idle time.

In this paper, we present several genetic algorithms, and analyse their performance on a wide range of instances. The proposed genetic approach uses a random numbers. The various versions of the genetic approach differ on the generation of the initial population, as well as on the use of local search. The genetic algorithms are compared with the best existing heuristic, as well as with optimal solutions for some instance size.

The remainder of this paper is organized as follows. In section 2, wedescribe our scheduling problem.In section 3, we describe theproposed memetic algorithm and geneticalgorithm approach, and presentthe several versions that were considered. Thecomputational results are reported in section 4. Finally, some concluding remarks are given in section 5.

2. Minimizing Total Linear Earliness and Quadratic Tardiness Costs

Our scheduling problem can describe as follows, for a survey see Baker and Scudder (1990):

A set of n jobs $N = \{1, 2, \dots, n\}$ are available for processing at time zero and each job j requires processing during an uninterrupted period of given length p_j, and ideally should be completed at its due-date d_j. Given a scheduling $(1, 2, \dots, n)$, then for each job j we calculate the completion time $C_j = \sum_{k=1}^{j} p_k$ such that no two jobs overlap in their execution, the earliness and tardiness of job j are defined by $E_j = max\{d_j - C_j, 0\}$ and $T_j = max\{C_j - d_j, 0\}$; correspondingly, a job is called early if it is completed before its due-date and tardy if it is completed after its due-date. If a job is completed exactly at its due-date, then it is called just-in-time. If schedule s is given then the quality of s is measured by the objective function $Z(s) = \sum_{j=1}^{n}(E_j + T_j^2)$.

To state our scheduling problem more precisely, we are given a set of n jobs are numbered $1, 2, \dots, n$. The objective is to find a processing order of jobs s, which minimizes the multiple objective functions (MOF) defined by:

$$\left.\begin{array}{lr} Min\, Z(s) = \underset{s\in S}{Min}\left\{\sum_{j=1}^{n}\left(E_{s_{(j)}} + T_{s_{(j)}}^2\right)\right\} & \\ s.t. & \\ C_{s_{(j)}} \geq p_{s_{(j)}} & j = 1, \dots, n \\ C_{s_{(j)}} = C_{s_{(j-1)}} + p_{s_{(j)}} & j = 2, \dots, n \\ E_{s_{(j)}} \geq d_{s_{(j)}} - C_{s_{(j)}} & j = 1, \dots, n \\ E_{s_{(j)}} \geq 0 & j = 1, \dots, n \\ T_{s(j)}^2 \geq (C_{s_{(j)}} - d_{s_{(j)}})^2 & j = 1, \dots, n \\ T_{s(j)}^2 \geq 0 & j = 1, \dots, n \end{array}\right\} (GP)$$

Where S is the set of all feasible solution, s is a schedule in$S$. The objective $Z(s)$ can be written as:

$$Z\left(s_{(j)}\right) = \begin{cases} d_{s_{(j)}} - C_{s_{(j)}} & if \quad C_{s_{(j)}} < d_{s_{(j)}} \\ (C_{s_{(j)}} - d_{s_{(j)}})^2 & if \quad C_{s_{(j)}} > d_{s_{(j)}} \\ 0 & if \quad C_{s_{(j)}} = d_{s_{(j)}} \end{cases}$$

Since the third term 0 when $C_{s_{(j)}} = d_{s_{(j)}}$ we can unhand it and put its values in any first or second terms because both terms took zero when $C_{s_{(j)}} = d_{s_{(j)}}$.This means that the cost of scheduling job $s_{(j)}$is $Z\left(s_{(j)}\right)$, given by:

$$Z\left(s_{(j)}\right) = \begin{cases} d_{s_{(j)}} - C_{s_{(j)}} & if \quad C_{s_{(j)}} \leq d_{s_{(j)}} \\ (C_{s_{(j)}} - d_{s_{(j)}})^2 & if \quad C_{s_{(j)}} > d_{s_{(j)}} \end{cases}$$

3. Memetic and Genetic Algorithms

GAs got recognition in the mid 70´s after John Holland's book entitled 'Adaptation in Natural and Artificial Systems' was published. Since then, due to its simplicity and efficiency, GAs became widespread, and in the 80´s, a new class of 'knowledge-augmented GAs', sometimes called 'hybrid GAs', started to appear in the literature. Recognizing important differences and similarities with other population-based approaches, some of them were categorized as memetic algorithms (MAs) in 1989 [10].

3.1 Population Structure

A population structure approach based on a ternary tree was chosen. In contrast with a non-structured population it divides the individuals in clusters and restricts crossover possibilities.

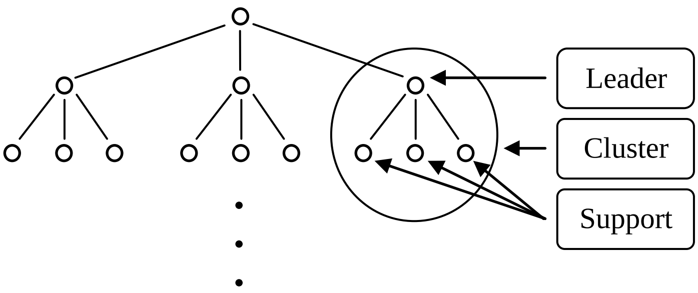

Figure 1. Population structure

The structure consists of several clusters and each cluster consists of a leader and three supporter solutions. The leader is chosen as the best individual of the cluster. The number of individuals in the population is defined by the number of nodes in the ternary tree, i.e., it is necessary 13 individuals to make a ternary tree with 3 levels, 40 individuals to 4 levels and so on.

3.2 Representation

For the SMS problem the representation we have chosen is quite intuitive, with a solution being represented as a chromosome with the alleles assuming different integer values in the [1, n] interval, where n is the number of jobs.

3.3 Crossover

Two different crossover operators were implemented. The first is the well-known Order Crossover (OX) [10]. After choosing two parents, a fragment of the chromosome from one of them is randomly selected and copied into the offspring. In the second phase, the offspring's empty positions are sequentially filled according to the chromosome of the other parent.

The second crossover calls homogeneous mixture crossover (HMX)was proposed by Mohammed [9], given by the mixture the two chromosomes from parents uniformly by make a set from genes M, he introduced the way for the mixture, first; the odd position from the first parent and the even position from the second parent. Then separate genes without repetition gene, since we read the set M from the left, if the gene j does not existing in the first child put it, otherwise we put gene j in the second child until final M. This way also gives a new two.

3.4 Mutation

In our implementation a traditional mutation strategy based on jobs swapping was implemented. According to it, two positions are randomly selected and the alleles in these positions swap their values.

3.5 Fitness Function

As in this problem the goal is to minimize the single machine scheduling problem with linear earliness and quadratic tardiness costs, the fitness function was chosen as randomly.

3.6 Offspring Insertion in Population

Once the leader and one supporter are selected, the recombination, mutation and local search take place and an offspring is generated. If the fitness of the offspring is better than the leader, the new individual takes its place. Otherwise it takes the place of the supporter that took part in the recombination. If the new individual is already present in the population, it is not inserted. We adopted a policy of not allowing duplicated individuals to reduce loss of diversity. After all individuals were inserted, the population is restructured. The fitness of the leader of a group must be lower than the fitness of the leader of the group just above it. Following this policy, the higher subgroups will have leaders with better fitness than the lower groups and the best solution will be the leader of the root subgroup. The adjustment is made by comparing the leader of each subgroup with the leader of the subgroup just above. If the leader in the level below turns out to be better, they swap their places.

4. Computational Experience

4.1 Test Problems

The GA was tested by coding it in Matlab 7.9 and running on a Pentium IV at 2.2GHz, with Ram 2GB computer.

The tested problem instances are generated as follows. For *n* = 10, 20, 30, 40, 50, 75, 100, 150, 200, 500,1000 and 2000, coefficient p_j for $j \in N = \{1, 2, \dots, n\}$ is generated by randomly selecting integers from interval [1,10]. It has been observed in the literature (e.g. [9]) that problem hardness is related to two parameters RDD and LF, called the relative range of due dates and the average lateness factor, respectively. In our experiment,

RDD = 0.2, 0.4, 0.6, 0.8, 1.0,

LF = 0.2, 0.4,

are used. Corresponding to each of these 5 × 2 = 10 cases, one problem instance is generated by selecting integer due dates $d_j, j \in N$, from interval

[(1 – LF – RDD / 2) *SP*, (1 – LF + RDD / 2) *SP*],

where $SP = \sum_{j \in N} p_j$. Sizes *n* = 10, 20, 30, 40, 50, 75, 100, 150, 200, 500, 1000 and 2000 are chosen.

4.2 Computational Results

In this section we will report on the results of our computational tests to show the effectiveness of our memetic algorithm (MA) and genetic algorithm (GA) methods. In table (1) we are going to compare between the results which obtain from the problem $1 \parallel \sum_{j=1}^{n} (E_j + T_j^2)$, it is clear from table (1) OX appear once one with GA in*n* = 30 and HMX giveagood solutions with MA and GA, since HMX with GA gives a good solutions with small and medium testing and HMX with MA has better with large testing.

Table 1: Mean results for MA, GA

	Memetic algorithm		Genetic algorithm	
n	HMX	OX	HMX	OX
10	164.7	164.3	164.5	166.6
20	1555.7	1591	1563.2	1566.4
30	2422.4	2427.4	2438.8	2416.5
40	5150.2	5282.2	5000.4	5191.6
50	8566.6	8742.9	8345	8660.5
75	29080.1	32243.4	27989.5	31116.9
100	59234.3	66648.4	56851.5	76202.4
150	235504.4	309623.4	243428.2	298422.7
200	764238.9	960344.3	719559.4	1012660.7
500	21108216.3	23834831.1	21630788.7	24637004
1000	214185569	243520187.2	225787147.2	236321386.2
2000	2115110240	2098478829	2137561277	2162864951

Table (2) shows that HMX with GA took best times for all iterations. And OX was worse with MA and GA

Table 2: Mean times for MA, GA

	Memetic algorithm		Genetic algorithm	
n	HMX	OX	HMX	OX
10	0.073306	0.1220918	0.0517894	0.1200178
20	0.1391599	0.185777	0.0863229	0.1794653
30	0.2082309	0.2734277	0.1999993	0.246813
40	0.2763997	0.3457631	0.2222919	0.3148616
50	0.3521817	0.4225856	0.2921941	0.3825081
75	0.5304999	0.6275855	0.437023	0.5583012
100	0.7060245	0.7153075	0.6676753	0.7432684
150	1.0560387	1.2661705	0.901375	1.1279571
200	1.4711303	1.7469457	1.2856587	1.5666435
500	5.1874992	5.5231803	4.1781054	5.1114735
1000	15.59945	15.030867	12.592579	14.258588
2000	48.695259	42.133792	39.575722	43.369643

5. Conclusion

In this paper, we considered the single machine scheduling problem with linear earliness and quadratic tardiness costs, and no machine idle time. Several heuristics based on the memetic algorithm and genetic algorithm approaches were presented. Results on GA and MA indicate the HMX as the best crossover operator. The OX crossover performed poorly, probably because of the quick loss of diversity in this crossover.And the HMX performed strongly because the approach mixture the chromosomes.

The main conclusion to be drawn from our computation results is that (HMX) is effective method for our problem especially for the large problem instances.

6. References

[1] Baker, K. R. and G. D. Scudder (1990), *Sequencing with earliness and tardiness penalties: A review*, Operations Research, 38, 22-36.

[2] Garey, M. R., R. E. Tarjan and G. T.Wilfong (1988), *One-processor scheduling with symmetric earliness and tardiness penalties*, Mathematics of Operations Research, 13, 330-348.

[3] Gupta, S. K. and T. Sen (1983), *Minimizing a quadratic function of job lateness on a single machine*, Engineering Costs and Production Economics, 7, 187-194.

[4] Hoogeveen, H. (2005), *Multicriteria scheduling*, European Journal of Operational Research, 167, 592-623.

[5] Kanet, J. J. and V. Sridharan (2000), *Scheduling with inserted idle time: Problem taxonomy and literature review*, Operations Research, 48, 99-110.

[6] Kim, Y. D. and C. A. Yano (1994), *Minimizing mean tardiness and earliness in single-machine scheduling problems with unequal due dates*, Naval Research Logistics, 41, 913-933.

[7] Korman, K. (1994), *A pressing matter*, Video, 46-50.

[8] Landis, K. (1993), *Group technology and cellular manufacturing in the Westvaco Los Angeles VH department*, Project report in IOM 581, School ofBusiness, University of Southern California.

[9] Mohammed H. A., (to appear): *Genetic algorithms for solving multiple objective function*. Iraqi Journal for Administrative Sciences.

[10] Moscato, P., *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*, Caltech Concurrent Computation Program, C3P Report 826 (1989).

[11] Schaller, J. (2002), *Minimizing the sum of squares lateness on a single machine*, European Journal of Operational Research, 143, 64-79.

[12] Schaller, J. (2004), *Single machine scheduling with early and quadratic tardy penalties*, Computers &Industrial Engineering, 46, 511-532.

[13] Schaller, J. (2007), *A comparison of lower bounds for the single-machine early/tardy problem*, Computers & Operations Research, 34, 2279-2292.

[14] Sen, T., P. Dileepan and M. R. Lind (1995), *Minimizing a weighted quadratic function of job lateness in the single machine system*, International Journal of Production Economics, 42, 237-243.

[15] Su, L.-H. and P.-C. Chang (1998), *A heuristic to minimize a quadratic function of job lateness on a single machine*, International Journal of Production Economics, 55, 169-175.

[16] Sun, X., J. S. Noble and C. M. Klein (1999), *Single-machine scheduling with sequence dependent setup to minimize total weighted squared tardiness*, IIE Transactions, 31, 113-124.

[17] Taguchi, G. (1986), *Introduction to Quality Engineering*, Asian Productivity Organization, Tokyo, Japan.

[18] Valente, J. M. S. (2006), *Heuristics for the single machine scheduling problem with early and quadratic tardy penalties*, Working Paper 234, Faculdade de Economia, Universidade do Porto, Portugal (to appear in European Journal of Industrial Engineering).

[19] Valente, J. M. S. (to appear), *An exact approach for the single machine scheduling problem with linear early and quadratic tardy penalties*, Asia-Pacific Journal of Operational Research, to appear.

[20] Wagner, B. J., D. J. Davis and H. Kher (2002), *The production of several items in a single facility with linearly changing demand rates*, Decision Sciences, 33, 317-346.