Australian Journal of Basic and Applied Sciences, 4(11): 5564-5570, 2010
ISSN 1991-8178

A Genetic Algorithm for Solving a New Mathematical Model of Single Machine Scheduling with Three Criteria

Mohammad Akhshabi

Department of Industrial Engineering, Islamic Azad University Tonekabon Branch , P.O. Box 46819-66856, Tonekabon, Iran.

Abstract: Appropriate programming in a gorge of plant or the key and strategic machine has much effect to increasing efficiency. Due to increasing global market competitiveness the regarded targets have become complex. Thus one criterion is not enough and scheduling with multiple criteria is more realistic. The main difficulty of these scheduling problems is extensive solving time needed to it. In this paper a new mathematical model with three criteria that contains customer satisfying and production costs is offered to single machine scheduling and solved with lingo8 software. Due to being NP-hard, Genetic Algorithm is developed which need less time than lingo8. Comparing of them show that lingo can not solve the instances with more than 10 jobs while Genetic Algorithm solve the instances with more than 100 jobs. The results show the most efficiency of Genetic Algorithms than lingo software.

Key world: Scheduling, Genetic Algorithms, Multi Criteria, Completion Time, Earliness Penalty, Tardiness Penalty, Single Machine

INTRODUCTION

In just-in-time (JIT) environment, each job should be completed as close as possible to its due date. It is involved producing goods only when necessary. Owning to the wide adoption of this philosophy in recent decades, scheduling problems for meeting the due date requirement have been investigated extensively, including those with general earliness-tardiness penalties about a common due date. Missing a Job's due date may result in loss of customer or the need to compensate for the delay along the production or assembly line. On the other hand, producing a job much earlier than its due date may cause unwanted inventory and/or deterioration of the product. in the modern competitive environment the cost of tardy deliveries, such as a company's goodwill, future sale and rush shipping cost, and the cost of early include holding cost for finished goods, deterioration of perishable goods and opportunity cost will significantly decrease a company profits. Therefore minimizing total weighted completion time, tardiness and earliness is not only a measure of academic interest but also useful and important in practice. Indeed flow shop, flexible flow shop, job shop and open shop scheduling problem are often addressed decomposing the original planning process into many sub problem that can be solved by using single machine techniques.

Most scheduling methods reported in the literature usually address the single or bi_criteria Scheduling problem and real single machine scheduling on the other hand, are often scheduled in relation on multi_criteria.

We first review several studies on the scheduling problems, followed by multicriteria (Two or more) scheduling problem and then we introduce genetic alghoritm and present our proposed scheduling problem with three criteria: total weighted completion time, earliness and tardiness penalty.

Some specific examples of production settings with these characteristics are provided by Ow and Morton (1989), Azizoglu *et al*. (1991), Wu *et al*. (1993) and Su and Chang (1988, 2001). The set of jobs is assumed to be ready for processing at the beginning which is a characteristic of the deterministic problem. As a generalization of weighted tardiness scheduling, the problem is strongly NP-hard in Lenstra *et al*. (1977). To the best of our knowledge, the earlier work in this problem is due to Chang and Lee (1992, 1992), Wu *et al*. (1993), and Chang (1999). Belouadah *et al*. (1992) dealt with the similar problem however with a different objective in minimizing the total weighted completion time and the problem is the same as discussed in Hariri and Potts (1983). Kim and Yano (1994) discussed some properties of the optimal solution, and these properties

Corresponding Author: Mohammad Akhshabi, Department of Industrial Engineering, Islamic Azad University Tonekabon Branch , P.O. Box 46819-66856, Tonekabon, Iran,
E-mail: m.akhshabi@tonekaboniau.ac.ir

are used to develop both optimal and heuristic algorithms. Valente and Alves (2003) presented a branch-and-bound algorithm based on a decomposition of the problem into weighted earliness and weighted tardiness subproblems. Two lower bound procedures were presented for each subproblem, and the lower bound for the original problem is then simply the sum of the lower bounds for the two subproblems. In Valente and Alves (2003), they analyzed the performance of various heuristic procedures, including dispatch rules, a greedy procedure and a decision theory search heuristic.

The early/tardy problem with equal release dates and no idle time, however, has been considered by several authors, and both exact and heuristic approaches have been proposed. Among the exact approaches, branchand-bound algorithms were presented by Abdul-Razaq and Potts (1988), Li (1988) and Liaw (1999). The lower bounding procedure of Abdul-Razaq and Potts was based on the subgradient optimization approach and the dynamic programming state-space relaxation technique, while Li and Liaw used Lagrangean relaxation and the multiplier adjustment method. Among the heuristics, Ow and Morton (1988) developed several dispatch rules and a filtered beam search procedure. Valente and Alves (2003) presented an additional dispatch rule and a greedy procedure, and also considered the use of dominance rules to further improve the schedule obtained by the heuristics. A neighborhood search algorithm was also presented by Li (1988). Genetic algorithm is a well-known technique and is used for many combinatorial optimization problems as in Holland (1975), Goldberg (1989) and David (1991). A good discussion of using genetic algorithms to problems that are encountered in production systems and operations research areas are available in Michalewicz (1994). Many researchers Chang *et al*. (2003, 2005, 2006) started using genetic algorithms for scheduling problems and a survey of genetic algorithms for job-shop scheduling is given in Chang *et al*. (1996).

seem's that any one do not consider total weighted completion time, earliness and tardiness penalty together.That a combined of these criteria which would answer customer needs and satisfaction.

2. Mechanics of Genetic Algorithm:

The basic procedure of a GA is described earlier. In the following sections we describe each of the GA mechanisms for our scheduling problem briefly.

2.1. Initialization and Evaluation:

A GA must be initialized with a starting population. The methods for creating an initial population are varied: feasible only, randomized, using heuristics, etc.

2.2. Reproduction:

Once we have an initial population, it is evaluated and noted as the parent population. Three methods of reproduction create the next generation (children): (1) elite, (2) mutation, and (3) crossover.

2.2.1. Elite:

The best solution or solutions found should be considered in subsequent generations. At a minimum, the single best solution from the parent generation needs to be copied to the next generation thus ensuring the best score of the next generation is at least as good as the prior generation. Here elite is expressed as a percent. For example, elite 1% means that we clone the top 1% of the population solutions for the next generation.

2.2.2. Mutation:

Mutation is expressed as a percent. For each solution in the parent population a random number is picked giving this solution a percent chance of being mutated. If this solution is picked for mutation then a copy of the solution is made and operation sequence steps inverted. Only operations from different jobs will be inverted so that the resulting mutation will always still be a feasible sequence. Mutation can be thought of as asexual reproduction.

2.2.3. Crossover:

Crossover is the breeding of two parents to produce a single child. That child has features from both parents and thus may be better or worse than either parent according to the objective function. Analogous to natural selection, the more fit the parent is the more likely they are to have offspring. A simple way of accomplishing this selection is via tournament selection. Two candidate solutions are randomly selected from the complete population, including those chosen as elite. The better ranked (in terms of score) of the two is selected to be parent 1. Then two more candidate solutions are randomly selected. The better scored of these last two candidates becomes parent 2. By using tournament selection, the more fit solutions get picked more often and the worst ranked less often. An alternate but equivalent method is to create a ‘‘roulette wheel” (Goldberg, D., 1989). The wheel has different width spaces so that the worst solution has the narrowest wedge increasing up to the best solution with the widest wedge.

By selecting operations in sequence from each parent, we ensure that the resulting child sequence is feasible. Sequences are turned into schedules via a simple earliest available or schedulable operation first rule. These simple rules guarantee that all child solutions are feasible and non-delay, as their parents were.

2.3. Generations:

After generating the initial population, the reproduction phase commences using the initial population members as parents to produce the first generation. Since the elite portion of reproduction clones the best scoring solution in the parent generation, we are assured that the next generation will have at least one member as good as the best from the prior generation. Therefore, scores can only stay the same or improve with each new generation. It logically follows that the more generations that are produced, the routine has more chance of finding a solution closer to the optimal one. The number of generations linearly impacts the running time of GA (Manikas, A. and Y.L. Chang, 2005).

2.4. Population Size:

We generate a random initial population according to the method described in Section 2.1.The larger the initial population that is created, the more likely the best solution from it will be closer to optimal.

For each subsequent generation, a large population size is also beneficial. For an equivalent mutate percent, there is a higher chance of a beneficial mutation when more mutations can occur because the population size increases. The amount of crossover breeding also increases as the population size goes up. The population size impacts the running time of GA at a near linear rate (Michalewicz, Z., 1994).

3. Mathematical Model:

i,j: Index of job

k, k' : Index of sequence

p_i : The needed time for performance of ith job

α_i : Penalty of earliness for ith job per hour.

β_i : Penalty of tardiness for ith job per hour.

θ_i : Weight of earliness for ith job *(i=1,2,….,N)*

η_i : Weight of tardiness for ith job *(i=1,2,….,N)*

γ_i : Weight of completion time for ith job *(i=1,2,….,N)*

$$\theta_i + \eta_i + \gamma_i = 1; \quad 0 < \theta_i < 1, \; 0 < \eta_i < 1, \; 0 < \gamma_i < 1,$$

λ_i : Weight of each criteria against other criteria in objective function (i=1,2,….,N)

$$xik = \begin{cases} 1 & \text{if ith job assign kth sequence.} \\ 0 & \text{otherwise} \end{cases}$$

$$\min\ z=\lambda_1\sum_{i=1}^{I}\ \theta_i\left(\left(\alpha_i\times\max(0,(d_i-c_i))\right)/\sum_{i=1}^{I}\alpha_i\times\max(0,(d_i-c_i))\right)$$

$$+\lambda_2\sum_{i=1}^{I}\ \eta_i\left(\left(\beta_i\times\max(0,(c_i-d_i))\right)/\sum_{i=1}^{I}\beta_i\times\max(0,(c_i-d_i))\right)\quad+\lambda_3\sum_{i=1}^{I}\gamma_i(c_i/\sum_{i=1}^{I}c_i)$$

s.t.:

$$\sum_{k=1}^{N}x_{ik}=1 \qquad i=1,...,N \tag{1}$$

$$\sum_{i=1}^{N}x_{ik}=1 \qquad k=1,...,N \tag{2}$$

$$\sum_{k=1}^{N}(x_{ik}(p_i+\sum_{k'=1}^{k-1}\sum_{\substack{i=1\\ i\neq j}}^{N}x_{ik'}p_i))=c_i \qquad i=1,...,N \tag{3}$$

$$x_{ik}\in\{0,1\} \quad i=1,...,N;\ k=0,1,...,N \tag{4}$$

when job i is completed earlier than the common due date, the earliness can be calculated by max (d_i - c_i) when multiplied in α_i shows the earliness penalty of ith order and divided by the sum of earliness penalty of jobs, it is transferred into standard variable and then by multipling in weight of earliness criteria against other criteria, the objective function of earliness is obtained. Similarly, the objective function of tardiness penalty, is obtained when job i is completed tardily than the common due date, the tardiness can be calculated by max(0, c_i - d_i) when multiplied in β_i shows the tardiness penalty of ith order and divided by the sum of tardiness penalty of jobs, it is transferred into standard variable and then by multipling in weight of tardiness criteria against other criteria, the objective function of tardiness is obtained. And third part of objective function shows the Completion Time.

The constraint (1) assigns per job one sequence. The constraint (2) assigns per sequence one job. The constraint (3) calculates the completion time of each job. The constraint (4) shows that x_{ik} is a binary variable.

4. Calculating Results:

To study the function of Genetic Algorithm, some example questions are randomly created and then the results obtained from the calculation of presented mathematical model by Lingo 8 software are compared with the calculation of the question by GA and are analyzed.

To calculate the above model, a PC with the specification of CPU 2.4 GHZ, 512 MB of RAM is used and GA algorithm has been expanded by MATLAB 7.0. To create there supposed questions of which the whole functions are at present zero time and the processing time of the works have been created randomly from the event distribution of U [10, 20] and tardiness penalty is created randomly between 3 till 8 and then multiplied by tardiness rate.

Since the tardiness penalty is more than the earliness penalty for each order a random number is produced from the event distribution of [1.5, 3] and multiplied by the earliness rate and as a result the amount of delay for the order is obtained. To calculate the delivery time of the jobs as applied in various articles and references regarding single machine questions , the delivery time is equal to d=pi*N(1-T) in which T is randomly obtain from U[0.1,0.7] for each N . The amount of λ1 is calculated by the following formula that λ1 is a random number.

$$\lambda_i = \frac{\lambda'_i}{\sum_j \lambda'_j}$$

Table 1: Comparing lingo8 and GA

Objective value		Solution time (second)		Job number	NUMBER
GA	Lingo	GA	Lingo		
.2628	.3172	.06	1	4	1
.2669	.3519	.17	8	5	2
.3018	.3628	.61	176	6	3
.3193	.3689	.92	9724	7	4
.3391	.3752	1.17	17138	8	5
.3392	-	1.15	>36000	10	6
.3425	-	1.70	-	15	7
.3508	-	2.33	-	20	8
.3685	-	3.02	-	25	9
.3756	-	3.81	-	50	10
.3889	-	4.52	-	75	11
.3978	-	5.97	-	100	12

In table (1) the results obtained from the Lingo8 software calculation and GA calculation for 5 to 100 dimension jobs have been compared.

In table (2) the mean of objective value that obtained from LINGO8 is 0.360017 and the mean of objective value that obtained from GA is 0.337767.

It seem's that the mean of LINGO objective value equals the mean of GA objective value.For exact statistical analysis, SPSS16 software was used and was calculated Independent Sample T Test, shown In table (3). As it is clear based on table(3), that significance value is 0.180 that means,equal variances assumed and throug exaiming we can consider that the significance value in first row i.e. 0.266 that shows the two means are equal.

Table 2: Group Statistics

	N	Mean	Std. Deviation	Std. Error Mean
LINGO	6	.360017	.0236489	
GA	12	.337767	.0437631	.0126333

Table 3: Independent Samples Test

	Levene's Test for Equality of Variances		t-test for Equality of Means						
	F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
								Lower	
Equal variances assumed	1.964	.180	1.152	16	.266	.0222500	.0193098	-.0186850	.0631850
Equal variances not assumed			1.399	1.577E1	.181	.0222500	.0159001	-.0114970	.0559970

As shown in the table (1), by the increase of the dimensions of the problem the difference between delivery time and the quality of the obtained result by the Lingo8 and GA will be increased. As specified, by the increase of the number of jobs, mathematical model is not able to provide result whereas in GA algorithm, the time of calculation is highly low. The quality of the result from GA is much better than Lingo8 software as well in Fig (1) and (2) The function of GA algorithm and Lingo8 are compared.

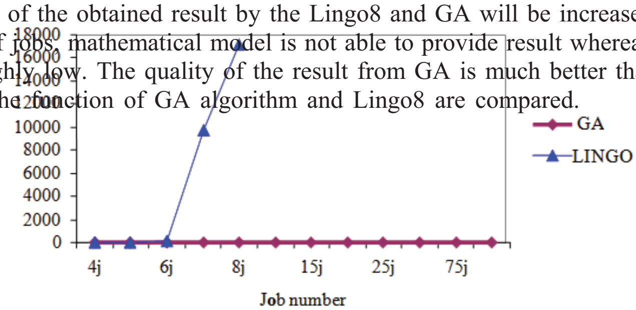

Fig. 1: Comparing of solution time GA and Lingo

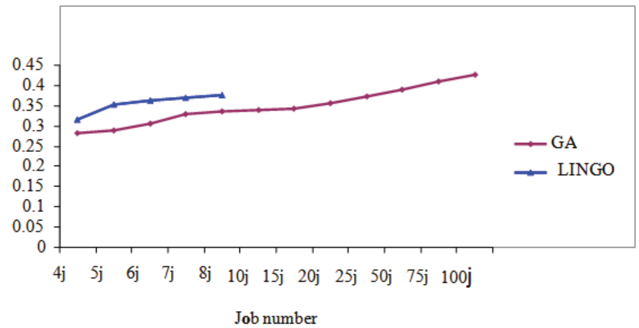

Fig. 2: Comparing of objective value GA and Lingo

5. Summary and Conclusion:

In this article, for more real scheduling, the question of just in time production has been considered with three criteria of earliness and tardiness penalty and completion time of jobs. To do so, new mathematical models are presented for the question. with regard to being NP-hard, the method of Genetic algorithm (GA) with the use of MATLAB 7.0 software has been developed and then the quality of the results with their time of calculation is compared with the results obtained from Lingo8 and analyzed and the efficiency of GA has been perfectly shown by the expansion of the said model for other state of production such as parallel machine series machine more researchers could done for future works.

Other Meta heuristic methods like Memetic algorithm, ant algorithm forbidden search algorithm could be used for the presented models as well.

ACKNOWLEDGMENTS

The author gratefully acknowledges the support provided by the management of the pars neopan company for their kind support and cooperation during field visits and data collection.

REFERENCE

Ow, P.S., E.T. Morton, 1989. The single machine early/tardy problem, Manage. Sci., 35: 171-191.

Azizoglu, M., S.M. Kondakci, O¨ mer Kirca, 1991. Bicriteria scheduling problem involving total tardiness and total earliness penalties, Int.Prod. Econom., 23(1-3): 17-24.

Wu, S.D., R.H. Storer, P.C. Chang, 1993. One machine heuristic with efficiency and stability as criteria, Comput. Operat. Res., 20.

Su, L.H., P.C. Chang, 1988. A heuristic to minimize a quadratic function of job lateness on a single machine, Int. J. Prod. Econom., 55: 169-175.

Su, L.H., P.C. Chang, 2001. Scheduling n jobs on one machine to minimize the maximum lateness with a minimum number of tardy jobs, Comput. Ind. Eng., 40: 349-360.

Lenstra, J.K., A.H.G. Rinnooy Kan, P. Brucker, 1977. Complexity of machine scheduling problems, Ann. Discret. Math, 1: 343-362.

Chang, P.C., H.C. Lee, 1992. A greedy heuristic for bi-criterion single machine scheduling problems, Comput. Ind. Eng., 22(2); 121-131.

Chang, P.C., H.C. Lee, 1992. A two phase approach for single machine scheduling: minimizing the total absolute deviation, J. Chin. Inst. Eng., 15(6): 735-742.

Chang, P.C., 1999. A branch and bound approach for single machine scheduling with earliness and tardiness penalties, Comput. Math. Appl., 37: 133-144.

Belouadah, H., M.E. Posner, C.N. Potts, 1992. Scheduling with release dates on a single machine to minimize total weighted completion time, Discret. Appl. Math, 36: 213-231.

Hariri, A.M.A., C.N. Potts, 1983. Scheduling with release dates on a single machine to minimize total weighted completion time, Discret.Appl. Math., 36: 99-109.

Kim, Y.D., C.A. Yano, 1994. Minimizing mean tardiness and earliness in single-machine scheduling problems with unequal due dates,Naval Res. Logist., 41: 913-933.

Valente, J.M.S., R.A.F.S. Alves, 2003. Heuristics for the early/tardy scheduling problem with release dates, Working Paper 129, Faculdadede Economia do Porto, Portugal.

Valente, J.M.S., R.A.F.S. Alves, 2003. Improved heuristics for the early/tardy scheduling problem with no idle time, Working Paper 126,Faculdade de Economia do Porto, Portugal.

Abdul-Razaq, T., C.N. Potts, 1988. Dynamic programming state-space relaxation for single machine scheduling, J. Operat. Res. Soc., 39: 141-152.

Li, G., 1988. Single machine earliness and tardiness scheduling, Eur. J. Operat. Res., 96: 46-558.

Liaw, C.F., 1999. A branch-and-bound algorithm for the single machine earliness and tardiness scheduling problem, Comput. Operat. Res., 26: 679-693.

Ow, P.S., T.E. Morton, 1988. Filtered beam searches in scheduling, Int. J. Prod. Res., 26: 35-62.

Holland, H.J., 1975. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, USA.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, New York, USA.

David, L., 1991. Handbook of Genetic Algorithms, Van Nostrand Reingold, New York, USA.

Michalewicz, Z., 1994. Genetic Algorithms + Data Structures = Evolution Programs, AI Series, Springer-Verlag, New York, USA.

Chang, P.C., J.C. Hsieh, Y.W. Wang, 2003. Genetic algorithms applied in BOPP film scheduling problems, Appl. Soft Comput., 3: 139-148.

Chang, P.C., S.H. Chen, K.L. Lin, 2005. Two phase sub-population genetic algorithm for parallel machine scheduling problem, Expert Syst. Appl., 29(3): 705-712.

Chang, P.C., J.C. Hsieh, C.H. Liu, 2006. A case-injected genetic algorithm for single machine scheduling problems with release time, Int. J.Prod. Econom., 103(2): 551-564.

Chang, R., M. Gen, T. Tsujimura, 1996. A tutorial survey of job shop scheduling problems using genetic algorithms , I. Representation,Comput. Ind. Eng., 30: 983-987.

Goldberg, D., 1989. Genetic algorithms in search, optimization, and machine learning. Indianapolis, IN: Addision Wesley.

Manikas, A. and Y.L. Chang, 2005. Using a genetic algorithm to solve the sequence dependent job shop scheduling problem. In Proceedings of INFORMS SE,Myrtle Beach, South Carolina.