Single Machine Scheduling against Restrictive Common Due Dates by Binding Constraints Approach

Eirini I. Nikolopoulou[1], George E. Manoussakis[2], George S. Androulakis[3]

[1]PhD Student, Department of Business Administration, University of Patras, Patras, Greece

[2] Special Staff, Department of Business Administration, University of Patras, Patras, Greece

[3] Associate Professor, Department of Business Administration, University of Patras, Patras, Greece

Abstract: *Scheduling problems involving earliness and tardiness penalties are critical for the operations management and in connection with just-in-time production and delivery, earliness as well as tardiness penalties are of interest. The objective of these problems is to minimize a summation of earliness and tardiness penalty costs. In this paper, a previous proposed linear programming method of identifying binding constraints is applied to find an initial solution of the single machine early/tardy scheduling problem against a restrictive common due date. This method uses a proposed notion: the average of each constraint, and is highly useful when dealing with large linear programming problems where only a relatively small percentage of constraints are binding at the optimal solution. The method identifies binding constraints in linear programming formulation of scheduling problems and more specifically, identifies whether a job is finished before (earliness), or after (tardiness) the specified due date. The knowledge of binding constraints, reduces the size of the problem and produces an initial solution which is close or identical to the optimal solution in problems where the optimal solution can be calculated. The method is applied on 400 instances of single machine scheduling problems in total ranging from 10 to 200 jobs to identify binding constraints and its performance is measured over 40 benchmark problems with 10 jobs taken from the open literature. The results are promising since the constraint identification was found able to propose solutions that are optimal or near optimal in the most restrictive against a common due date examined test problems. The method is also applied on test problems that optimal solutions can't be calculated leading to significant reduction of their size, which is useful in case of large scheduling problems that demand space.*

Keywords: *Job scheduling, Common due date, Binding constraints identification, Linear programming, Benchmarks*

1. INTRODUCTION

This paper deals with the single-machine early/tardy scheduling problem of a set of jobs with a restrictive common due date and objective the minimization of the jobs' total earliness and tardiness. When scheduling against a common due date some of the jobs may be completed early (prior to the due date), while others are finished late. In both cases costs are incurred; early jobs cause holding costs while tardy jobs could result to penalties such as loss of customer goodwill, loss of reputation and loss of orders. Therefore, earliness as well as tardiness of jobs should be discouraged.

Common due-date problems have been studied extensively during the last years and many approximate algorithms have been proposed for various versions of the basic problem. Since Kanet's [1] work, many researchers have extended this contribution proposing new algorithms [2-11]. The majority of the proposed algorithms for addressed instances of the problem with a small number of jobs, up to 25 or 50 jobs. Recently, the use of meta-heuristics enabled researchers to address these problems more effectively [12-17]. Moreover, Biskup & Feldman [18] generated a set of benchmarks for single-machine early/tardy scheduling problems together with their upper bounds on the optimal objective functions.

A major objective of this work is to show how a proposed linear programming method for identifying binding constraints can be applied with success on these problems. The underlying idea, is to use the identified binding constraints of common due date problems in linear programming form to obtain an optimal or a near optimal solution. The application in 10 jobs problems was promising, and the results could be considered as an initial solution and used combined with other methods in problems with more jobs.

The contribution of this method, is to produce an initial solution based on the binding constraints of the linear programming formulation. This method cannot replace the mentioned methods, however it may improve them by providing a feasible starting solution even in large scale problems, using only a relatively small amount of data of the problem. The experiments regarding the optimal solution have been carried out over the test instances of benchmarks for 10 jobs [18]. The application of the method in problems up to 200 jobs, identifies the binding constraints that lead to a near optimal solution. Since many scheduling problems are actually large scaled, the development of techniques for reducing considerably their dimension of the problem, is an inevitable need. This problem reduction, among others, results to less computational time and effort.

This paper is organized as follows: Section 2 states the problem, Section 3 gives a description of the proposed algorithm for identifying binding constraints and Section 4 presents and discusses the results of the experimental evaluations of the algorithm. Finally, Section 5 summarizes the contribution of the paper and states some directions for future work.

2. PROBLEM FORMULATION

2.1 Problem representation

Consider n jobs available at time zero, which have to be processed on a single machine. Each of these jobs needs exactly one operation. The processing times p_j of the jobs $j = 1,2,\dots,n$ are deterministic and known and preemption of jobs is not allowed. If the completion time C_j of job j is smaller than or equal to the common due date d, which is assumed as given, the jobs' earliness is $E_j = d - C_j$. Accordingly, a job j is tardy with the tardiness $T_j = C_j - d$, if its completion time is greater than the common due date d As it is not known in advance whether a job will be completed before or after the due date, earliness and tardiness are calculated as:

$E_j = \max(0, d - C_j)$

and $T_j = \max(0, C_j - d)$ for all jobs $j = 1,2,\dots,n$.

The per time unit penalties of the job j for being early or tardy are a_j and β_j, respectively. Penalties are incurred whenever a job is completed before or after this due date. Therefore, an ideal schedule is one in which all jobs finish on the specific due date.

The objective is to jointly minimize the sum of earliness and tardiness penalties $f(S) = \sum_{j=1}^{n}(a_j E_j + \beta_j T_j)$ (1) where S denotes a feasible schedule of the jobs.

In objective function (1), the common due date d might be a decision variable whose value has to be determined, or it might have been given externally. Suppose a due date $d > \sum_{j=1}^{n} p_j$ is given; an (global) optimal schedule S with the sequence of the jobs around this due date and its objective function value f(S) can be constructed. A common due date is called unrestrictive as long as the optimal schedule S (with f(S)) can be realized. Obviously, an (externally) given common due date, for which $d \geq \sum_{j=1}^{n} p_j$ is unrestrictive. Furthermore, the due date is called unrestrictive if it is a decision variable. To summarize, we refer to a common due date as unrestrictive, if the optimal sequence can be constructed without considering the (value of) the due date. Otherwise the common due date is called restrictive. Consequently, a problem can be considered as either unrestricted or restricted. The restricted version of the problem is obviously more difficult.

The basic assumptions in restricted common due sate problems can be summarized as follows:

- Machine breakdown and maintenance are neglected
- The machine is continuously available and never kept idle while there are jobs waiting to proceed
- The machine processes only one job at a time
- Jobs are known in advance
- Jobs are independent without precedence or other constraints.

Furthermore, for restricted common due sate problem with general earliness and tardiness penalties there is an optimal schedule with the following properties:

1) No idle times are inserted between consecutive jobs
2) The schedule is V-shaped: the jobs j which are completed at or before the due date $C_j \leq d$ are ordered according to non-increasing ratios $\frac{p_j}{\alpha_j}$, $j = 1,2,\dots,n$ while the jobs j whose processing starts at or after the due date $C_j - p_j \geq d$ are sequenced in non-decreasing order of the ratios $\frac{p_j}{\beta_j}$, $j = 1,2,\dots,n$
3) The processing time of the first job either starts at time zero, or one job is completed at the due date.

The value of a restrictive factor $h =$ 0.2, 0.4, 0.6, 0.8 classifies the problems as less or more restricted against a common due date:

$$d = round[h \sum_{j=1}^{n} p_j]$$

where $round[h \sum_{j=1}^{n} p_j]$ denotes the biggest integer which is smaller than or equal to $h \sum_{j=1}^{n} p_j$.

2.2 Linear programming formulation

To obtain optimal schedules S for small instances of this problem the following mixed-integer programming formulation is applied.

Let s_j and x_{jk} the decision variables, that determine the schedule S; s_j is the corresponding variable to the starting time of job j and x_{jk}:

$$x_{jk} = \begin{cases} 1, if\ job\ j\ is\ sequenced\ prior\ to\ job\ k \\ 0, otherwise \end{cases}$$

A job can be sequenced, not necessarily directly, prior to another. Furthermore, let R be a large number.

The objective is to find a schedule which minimizes (1) subject to the following restrictions:

$$minimize\ f(S) = \sum_{j=1}^{n}(a_j E_j + \beta_j T_j)$$

subject to

$T_j \geq s_j + p_j - d,\ j = 1,2,\dots,n$ (2)

$E_j \geq d - s_j - p_j,\ j = 1,2,\dots,n$ (3)

$s_j + p_j \leq s_k + R(1 - x_{jk}),$

$j = 1,2,\dots,n-1,\ k = j+1,2,\dots,n$ (4)

$s_k + p_k \leq s_j + Rx_{jk},$

$j = 1,2,\dots,n-1,\ k = j+1,2,\dots,n$ (5)

$T_j \geq 0,\ j = 1,2,\dots,n$ (6)

$E_j \geq 0,\ j = 1,2,\dots,n$ (7)

$s_j \geq 0,\ j = 1,2,\dots,n$ (8)

$x_{jk} \in \{0,1\}, j = 1,2,\dots,n-1,\ k = j+1,2,\dots,n$ (9)

The values for tardiness and earliness are calculated by constraints (3) and (3). Constraints (4) and (5) determine the starting times of the jobs: If job j is sequenced prior to job k the constraint (4) only holds if$x_{jk} = 1$. Due to the addition of R (5) is not binding with$x_{jk} = 1$. On the other hand, for $x_{jk} = 0$, (4) is not binding. This formulation uses the order-dependent binary variables proposed by [19]. Thus, we have a minimization problem with a total of $2n$ variables and $4n - 2$ constraints without including in this number the positivity constraints and the binary variable constraints. Nevertheless, even in this formulation, the problem remains computationally demanding.

Due to the complexity results of [7] and [8] it is most unlikely to find efficient algorithms for the restricted common due date problem (1). Only for small problems with up to 10 jobs was an optimal schedule found in a reasonable time by applying the mixed-integer programming formulation given in (1)-(9) and using standard software (LINGO and LINDO). Therefore problems with more jobs are usually approached using metaheuristics [12-17].

3. A BINDING CONSTRAINTS ALGORITHM IMPLEMENTATION TO THE RESTRICTIVE GENERAL PROBLEM

For very large linear programming problems, only a relatively small percentage of constraints are binding at the optimal solution. In linear programming, a constraint is called binding if it is satisfied as an equality at the optimal solution, otherwise the constraint is called redundant. The proposed algorithm is a binding constraints identification method based on the weighted mean of the constraints of a linear programming problem, see [20]. The number of operations required in the proposed method is small compared to other known algorithms and its complexity is $O(mn^2)$, while no artificial variables are needed. The algorithm checks for binding constraints considering one decision variable each time. Using the weighted average and the intercepts of the constraints with the zero-level hyperplane of the variable under consideration, the algorithm moves from a constraint to an adjacent one until it locates a binding constraint. The method was applied in three sets of 1000 different random small, medium and large scale linear programming problems to check its efficiency in identifying binding constraints. Since there was no information about the constraints in these random problems, the proposed algorithm was considered as a statistical tool for correctly identifying binding constraints. The probability that binding constraints were identified correctly by the algorithm is 86.9%, 89% and 94.2% in small, medium and large scale problems respectively.

Given an instance with 8 jobs ($n = 8$) and the following data (Table 1), the optimal objective function value can be obtained by applying the above linear programming formulation (1)-(9).

In this problem, the total sum of the processing times of the jobs is $\sum_{j=1}^{8} p_j = 70$. For the restrictive problem with $d = 0.2 * \sum_{j=1}^{8} p_j = 14$, the optimal solution is equal to 1066. The method presented in [20] was applied in a problem with 8 jobs to identify the binding constraints that lead to the optimal solution.

Table1: Example instances for $n = 8$

	j_1	j_2	j_3	j_4	j_5	j_6	j_7	j_8
p_j	7	1	18	6	13	14	13	8
a_j	2	8	4	9	5	5	7	4
β_j	14	7	8	9	7	9	5	14
p_j/a_j	3.5	0.13	4.5	0.67	2.6	2.8	0.71	1.5
p_j/β_j	0.5	0.14	2.3	0.67	1.86	1.56	1	0.43

According to linear programming form (1)-(9), in this problem there are 30 constraints: constraints 1-8 refer to tardiness, constraints 9-16 refer to earliness, while constraints 17-23 and constraints 24-30 determine the starting times of the jobs. Applying the proposed algorithm, the identified binding constraints from different weighted means, are:

- 18, 1, 9, 17, 19, 5, 21, 22, 23, 30, 26
- 15, 16, 2, 1, 10, 18, 4, 5, 21, 7, 8, 30, 26
- 18, 9, 4, 30, 17, 12, 5, 21, 22, 23, 26
- 25, 5, 30, 26, 13, 28
- 16, 12, 7, 30, 17, 18, 19, 5, 21, 15, 23, 26
- 18, 9, 8, 30, 17, 19, 5, 21, 22, 16, 26

Depending on the frequency of the constraint identification, the fifth job should be considered tardy while the first and eighth jobs should be early. The second job is identified in the set of the constraints having the biggest length and can be characterized as either early or tardy. If this job is considered early, the optimal schedule is obtained. Early and tardy jobs are placed according to scheme V and the order of all early jobs is $S_E = \{1, 8, 2\}$ while for all tardy jobs, $S_T = \{4, 7, 6, 5, 3\}$. In this procedure, we focused on the tardiness and earliness constraints, while the constraints referring to the starting times of the jobs were used ancillary to the optimal schedule.

The results of the proposed algorithm show that we need just 13 binding constraints to find the solution. Therefore, the initial problem of the 30 constraints is reduced to a problem of 13 constraints which leads to the optimal solution. The reduction of the initial problem dimension is about 56.67%.

Table 2: The schedule for the n=8 problem instance obtained by the identification of binding constraints

			$d = 14$				
early			**tardy**				
j_1	j_8	j_2	j_4	j_7	j_6	j_5	j_3
7	6	1	6	5	14	13	18
0	13	14	20	25	39	52	70

4. EXPERIMENTAL ANALYSIS AND DISCUSSION

The full computational results obtained by the proposed algorithm concerning the 10 jobs benchmark problems are summarized in Table 3. The optimal objective function values for the instances with 10 jobs are given in square brackets, if they are not obtained by the algorithm, otherwise, they are given without brackets. The underlined results, are nearer to the optimal solution than the results proposed by [18]. As it was mentioned, optimal solutions for these benchmarks only exist for the 10 jobs test instances and have been achieved using an integer programming formulation with LINDO software αναφορά, therefore the comparative results refer only to these problems, for four restrictive factors. The algorithm was also applied on the benchmarks problems with jobs ranging from 20 to 200, indicated the reduction of the size of these problems. Table 4 contains the above results. Since the proposed algorithm was tested in problems up to 1000 constraints according to linear programming formulation, benchmarks problems with jobs ranging from 500 to 1000 were not tested.

Table 3: The optimal objective function values and the upper bounds for the 10 job examples

n=10	$\sum_{j=1}^{n} p_j$	h=0.2	h=0.4	h=0.6	h=0.8
k=1	116	<u>1944</u> [1936]	1025	876 [841]	818
k=2	129	<u>1107</u> [1125]	615	877 [615]	759 [615]
k=3	125	<u>1622</u> [1586]	940 [917]	906 [793]	869 [793]
k=4	102	<u>2153</u> [2139]	<u>1251</u> [1230]	815	906 [803]
k=5	94	<u>1194</u> [1187]	<u>649</u> [630]	563 [521]	726 [521]
k=6	88	<u>1572</u> [1521]	908	797 [755]	873 [755]
k=7	103	2170	1496 [1374]	1182 [1101]	1163 [1083]
k=8	79	1720	1106 [1020]	630 [610]	576 [540]
k=9	92	1574	884 [876]	582	592 [554]
k=10	127	1869	1136	710	780 [671]

The algorithm based on binding constraints identification, tends to find the optimal or near optimal solution in problems with the most restrictive due dates, while in problems with less restrictive dates tends to lead to higher upper bounds. In particular, for all the test problems with 10 jobs and h=0.2, the proposed algorithm leads to better or identical solution comparing to the results of [18]. Specifically, in 4 out of 10 problems the proposed solution is identical and in the other 6 problems the obtained solution is better. The average improvement of these solutions is 53.70%. In these problems, Biskup & Feldmann's algorithm [18] fails to find exactly the optimal solution. For problems with h=0.4 and h=0.6 the proposed algorithm leads to the optimal solution in 4 and 2 problems respectively, while for problems with h=0.8 the algorithm fails to find the optimal solution of the schedule. Conversely, Biskup & Feldmann's algorithm [18] succeeds in finding the optimal solution in problems with less restrictive due dates. The comparative results for h=0.2 are illustrated in Chart 1.

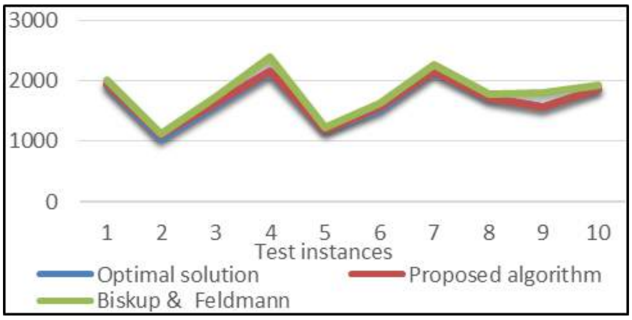

Chart -1: Comparing results for h=0.2

The application of the algorithm indicated that for the 10 instances of 10 jobs problem, the binding constraints demanded to find a solution were 13-16, 14-17, 13-17 and 14-17 for h = 0.2, 0.4, 0.6, 0.8 respectively while for 20 jobs, the binding constraints were 21-27, 21-26, 22-25 and 21-29. For 50, 100 and 200 jobs the binding constraints were 51-53, 103 and 203 respectively. For problems from 100-200 jobs, the number of binding constraints is quite the same for all factors h, however the identified constraints were different for each factor. Consequently, instead of the initial problem, we can use a reduced sized problem to product a starting feasible solution. The results of the reduction as the average percentages of the 10 instances for each job problem, are presented in Table 4. It is obvious, that for larger problems (up to 100-200 jobs) the reduction is greater than the reduction for small problems.

Table 4: Average reduction of the size of the linear programming formulation for the 10, 20, 50, 100 and 200 job examples

	h=0.2	h=0.4	h=0.6	h=0.8
n=10	61.32%	59.73%	61.32%	60.53%
n=20	68.74%	67.65%	70.38%	69/99%
n=50	73.63%	73.59%	73.64%	73.43%
n=100	74.12%	74.12%	74.12%	74.12%
n=200	74.56%	74.56%	74.56%	74.56%

5. CONCLUSIONS

In this paper an application of a proposed algorithm is presented for the restricted common due date problem. The method identifies binding constraints in linear programming problems, providing a starting feasible solution for the restricted common due date problem. As this problem is NP-hard the implementation of optimizing algorithms is difficult, however the application of the method in 10 jobs problems was promising. The results could be considered as starting solutions and used in conjunction with other methods in problems with more jobs to construct more powerful algorithms. Furthermore, by identifying the binding constraints, the dimension of large scaled scheduling problems is reduced. Thus, the application of the method results to less computational time and effort. The method is quite general and can be rather easily modified, improved and applied to any other scheduling problem in future.

REFERENCES

[1] J.J. Kanet, “Minimizing the average deviation of job completion times about a common due date”, Naval Research Logistics Quarterly, vol. 28, pp. 643-651, 1981.

[2] U. Bagchi, R.S. Sullivan, and Y.L. Chang, “Minimizing absolute and squared deviations of completion times with different earliness and tardiness penalties about a common due-date”, Naval Research Logistics Quarterly, vol. 34, pp. 739-751, 1987.

[3] T. Abdul-Razaq and C.N. Potts, “Dynamic programming state-space relaxation for single-machine scheduling”, Journal of the Operational Research Society, vol. 39, pp. 141-152, 1988.

[4] T. Cheng and M. Gupta, “Survey of scheduling research involving due date determination decision”, European Journal of Operational Research, vol. 38, pp. 156-166, 1989.

[5] K. Baker and G. Scudder, “Sequencing with earliness and tardiness penalties: A review”, Operations Researchm vol. 38, pp. 22-36, 1990.

[6] N. Hall and M. Posner, “Earliness-tardiness scheduling problems I: weighted deviation of completion times about a common date”, Operations Research, vol. 39, 836-846, 1991.

[7] N. Hall, G.W. Kubiak, and S.P. Sethi, “Earliness–tardiness scheduling problems, II: Deviation of completion times about a restrictive common due date”, Operations Research, vol. 39, pp. 847–856, 1991.

[8] J.A. Hoogeveen and S.L. van de Velde, “Scheduling around a small common due date”, European Journal of Operational Research, vol. 55, pp. 237-242, 1991.

[9] J.P. Souza and L.A. Wolsey, “A time indexed formulation of non-preemptive single machine scheduling problems”, Mathematical Programming, vol. 54, pp. 353-367, 1992.

[10] P. De, J.B Ghosh and C.E. Wells, “Solving a generalized model for CON due date assignment and sequencing”, Int. Journal of Production Economics vol. 34, pp.179-185, 1994.

[11] V. Gordon, J.-M.Proth, and C. Chu, “A survey of the state-of-the-art of common due date assignment and scheduling research”, European Journal of Operational Research, vol. 139, pp. 1-25, 2002.

[12] C.-Y. Lee and S.J. Kim, “Parallel genetic algorithms for the earliness-tardiness job scheduling problem with general penalty weights”, Computers and Industrial Engineering, vol. 28, pp. 231-248, 1995.

[13] Q. Hao, Z. Yang, D. Wang, Z. Li, “Common due date determination and sequencing using tabu search”, Computers and Operations Research, vol. 23, pp. 409-417, 1996.

[14] R.J.W. James, “Using tabu search to solve the common due date early/tardy machine scheduling problem”, Computers and Operations Research, vol. 24, pp. 199-208, 1997.

[15] M. Feldmann and D. Biskup, “Single-machine scheduling for minimizing earliness and tardiness penalties by metaheuristic approaches”, Computers and Industrial Engineering, vol. 44, pp. 307-323, 2003.

[16] A.C. Nearchou and S.L. Omirou, “Differential evolution for sequencing and scheduling optimization, Journal of Heuristics”, vol. 12, pp. 395–411, 2006.

[17] Z.-J. Lee, S.-W. Lin, K.-C. Ying, “A dynamical ant colony optimization with heuristics for scheduling jobs on a single machine with a common due date”, Studies in Computational Intelligence, vol. 128, pp. 91-103, 2008.

[18] D. Biskup and M. Feldmann, “Benchmarks for scheduling on a single machine against restrictive

and unrestrictive common due dates", Computers and Operations Research, vol. 28 pp. 787-801, 2001.

[19] A.S. Manne, "On the job-shop scheduling problem". Operations Research, vol.8, 219-223, 1960.

[20] E.I.Nikolopoulou, G.E. Manoussakis and G.S. Androulakis, "Locating binding constraints in LP Problems", American Journal of Operations Research, vol.09, No.02 Article ID:90969, 2019.