Journal of Quality Engineering and Production Optimization

Vol. 4, No. 1, PP. 17-24, 2019

Comparing Mixed-Integer and Constraint Programming for the No-Wait Flow Shop Problem with Due Date Constraints

Hamed Samarghandi*[1] and Farzad Firouzi Jahantigh[2]

Department of Finance and Management Science, Edwards School of Business, University of Saskatchewan, Saskatoon, Saskatchewan, Canada, S7N 5A7
Department of Industrial Engineering, University of Sistan and Baluchestan, Zahedan, Iran (farzadfirouzi@gmail.com)

***Corresponding Author:** *Hamed Samarghandi (E-mail: samarghandi@edwards.usask.cam)*

Abstract *– The impetus for this research was examining a flow shop problem in which tasks were expected to be successively carried out with no time interval (i.e., no wait time) between them. For this reason, they should be completed by specific dates or deadlines. In this regard, the efficiency of the models was evaluated based on makespan. To solve the NP-Hard problem, we developed two mathematical models. Once we solved our problem using Mixed-Integer Programming Model (henceforth MIPM) and then, we applied a Constraint Programming Model (CPM); finally, we compared the optimality of the presented results.*

Keywords *– Constraint programming model, Flow shop scheduling, Makespan, Mixed-integer programming model, Specific deadlines.*

I. INTRODUCTION

In this paper, we focus on no-wait flow shop scheduling problem in which tasks are successively operated. Moreover, there is no interval or interruption between these tasks and even the operations of the same task are not interrupted. The assumption is that there is a time constraint for each job in which it should be completed. We assume that the tasks need to be finished at time zero. This means the tasks are all set to the earliest time (henceforth the release date), which is zero; the optimality of models can be evaluated based on makespanc while the NP-hardness of $F \mid no-wait, d_i \mid C_{\max}$ can be approved (Samarghandi & ElMekkawy, 2012).

Industrial implications of $F \mid no-wait, d_i \mid C_{\max}$ have been underlined in the literature on different disciplines such as chemistry (Afshar-Nadjafi, 2014); nutrition (Hall & Sriskandarajah, 1996); manufacturing; e.g., steel (Wismer, 1972) or concrete (Raaymakers & Hoogeveen, 200); and pharmacy (Grabowski & Pempera, 2000). Specific time constraints have been addressed as hard constraints. The reason for the gap we find is that scholars have often developed a feasible model for the problem and then, have shown that the feasibility is a big challenge, especially when due dates are closer. The literature is full of proposed methods in which the time constraints are removed to address the no-wait scheduling problem.

A review of studies shows that scholars have drawn upon programming models to address ordering and scheduling issues. For instance, Selen & Hott (1986) used MIPM to schedule flow shop tasks and executed them to more than one apparatus or machine. In proposing a Mixed Integer Linear Programming Model (MILPM), Stafford (1988) used the model developed by Wagner (1959). In another empirical study, Tseng et al. (2004) assessed the efficiency of different MIPMs for permutation flow-shop scheduling problems; their results were quite in line with those reported by Pan (1997) for both work-shop and flow-shop problems. To have the best solution to the reentrant workshop scheduling problem, Pan & Chen (2005) adopted a Mixed Binary Integer Programming Model (MBIPM). In

Manuscript Received:06- Oct- 2018 and revised: 05-Dec-2018 ……………………………………………ISSN: 2423-3781
Accepted: 26-Feb-2019……………………………………………………………………Research Paper

the same vein, Ziaee & Sadjadi (2007) solved and compared 7 MBIP models for a problem in which flow-shop tasks were sequenced. In another study, using fuzzy objective functions, Javadi et al. (2008) solved a Linear Programming Model (LPM) for the flow shop problem with no elapsed time between the tasks. Ramezanian et al. (2010) developed a mathematical programming model to cut down the expenses of a flow shop context in which processing times were assumed zero.

Reviewing the available research (see Lee et al., 2014; Yazdani et al., 2010; Zandieh et al., 2006) shows that scheduling for an activity or operation has been carried out considering constant processing time. However, the processing time can be multimode. In fact, when more resources are assigned to an activity, the processing time is reduced. When there are constraints on the available resources, in addition to scheduling of the activities, the allocation of available resources to them needs to be considered.

The present research proposes two mathematical algorithms, namely a constraint programming model (CPM) and an MIPM, for $F \mid no-wait, d_i \mid C_{\max}$. Specific time constraints are considered as hard ones. Analysis of computational data in this research reveals that the decrease in the number of jobs in $F \mid no-wait, d_i \mid C_{\max}$ leads to achieving the best possible result for small problems. Such a reduction in jobs in a problem instance is more prominent when the due dates are closer.

II. PROBLEM DESCRIPTION

The assumptions for $F \mid no-wait, d_i \mid C_{\max}$ are as follows: 1) scheduling all jobs or tasks has to be done in a specific order; 2) interruption between the jobs is not allowed; 3) each task should be executed on an apparatus only once at a time and each machine should process a single operation once at a time; and 4) tasks should be successively operated with no wait time. We present our scheduling models using the following indices:

A	Set of apparatuses
a = |A|	Number of apparatuses
N	Set of tasks
$n = \|N\|$	Number of tasks
Ti	Task i
o_{ij}	j th operation of Ti
p_{ij}	Time required for the j th operation of Ti on a specific apparatus
Bi	Time at which Ti has begun
Boij	Time at which o_{ij} has begun
d_i	Specific date of Ti
π_l	Sequence or order of l
$C_{\max}$	Makespan of π_l

Brackets show that the tasks are scheduled successively. For example, Bi is defined as the time at which the task is expected to begin and needs to be scheduled after the i th task in a specific succession.

III. PROPOSED MODELS

A. Mixed-integer programming model

We defined the decision variable by constraint **Error! Reference source not found.**.

$$x_{ij} = \begin{cases} 1 \text{ if } J_j \text{ is placed immediately after } J_i \text{ in the sequence} \\ 0 \quad \text{Otherwise} \end{cases} \tag{1}$$

$i, j = 1, 2, ..., n$

Our algorithm would be

$$Min\ C_{\max} \tag{2}$$

$$C_{\max} \geq S_{o_{im}} + p_{im};\ i = 1, 2, ..., n \tag{3}$$

$$S_{o_{k1}} + M(1 - x_{ik}) \geq S_{o_{i1}} + p_{i1};\ i = 1, 2, ..., n;\ k = 1, 2, ..., n \tag{4}$$

$$S_{o_{i[j]}} = S_{o_{ij}} + p_{ij};\ i = 1, 2, ..., n;\ j = 1, 2, ..., m-1 \tag{5}$$

$$S_{o_{im}} + p_{im} \leq d_i;\ i = 1, 2, ..., n \tag{6}$$

$$\sum_{i=1}^{n} x_{ij} \leq 1;\ j = 1, 2, ..., n \tag{7}$$

$$\sum_{j=1}^{n} x_{ij} \leq 1;\ i = 1, 2, ..., n \tag{8}$$

$$x_{ij} + x_{ji} \leq 1;\ i = 1, 2, ..., n;\ j = 1, 2, ..., n \tag{9}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} = n - 1 \tag{10}$$

$$S_{o_{ij}} \geq 0;\ i = 1, 2, ..., n;\ j = 1, 2, ..., m \tag{11}$$

$$x_{ij} \in \{0, 1\};\ i = 1, 2, ..., n;\ j = 1, 2, ..., n \tag{12}$$

In this model, we attempt to decrease the makespan. A is a big number. According to constraint (3), the value of makespan is equal to the completion time of the previous tasks. Constraint (4) ensures that operations do not overlap and it is the binding provided that Tk is scheduled right away after Ti in the specific order. Constraint (5) enforces no-wait constraints. Constraint (6) determines the specific time constraint. Constraint (7) ensures that each task is over before the given time. Constraints (8), (9), (10), and (11) ensure that all the jobs are to be executed exactly once in the sequence determined.

B. Constraint programming model

In developing this **Error! Reference source not found.**, we considered features of constraint programming. We defined the decision variable of this model as follows:

$x_i = j$ if J_j is in the place of i.

Thus, our model would be:

$$\min\left(S_{o_{x_n,m}} + p_{x_n,m}\right) \quad (13)$$

$$\text{All Different}(x_1, x_2, ..., x_n) \quad (14)$$

$$S_{o_{x_{(i+1)},k}} \geq S_{o_{x_i,k}} + p_{x_i,k}; k = 1,2,...,m; i = 1,2,...,n-1 \quad (15)$$

$$S_{o_{x_i,[j]}} = S_{o_{x_i,j}} + p_{x_i,j}; j = 1,2,...,m-1; i = 1,2,...,n \quad (16)$$

$$S_{o_{x_i,m}} + p_{x_i,m} \leq d_{x_i}; i = 1,2,...,n \quad (17)$$

$$S_{o_{ij}} \geq 0; i = 1,2,...,n; j = 1,2,...,m \quad (18)$$

$$x_i \in \{1,2,...,n\}; i = 1,2,...,n \quad (19)$$

In this model, Constraint (15) ensures that the jobs should overlap each other. Constraint (16) enforces the no-wait constraints and Constraint (17) shows the due date constraints. Numerical findings are detailed in the following section.

C. COMPUTATIONAL ANALYSIS

To assess the optimality of our models, we carried out numerical experiments using IBM ILOG CPLEX V12.5. The computer used in this study was a 3GHz Intel Pentium IV CPU with 4 GB of RAM. To carry out the analysis, we reviewed the literature and chose 8 test problems for $F|no-wait|C_{\max}$; the car problems were derived from the study of Carlier (1978) (for more information and access to the test, refer to the OR-Library (Beasley, 1990)). The optimal solution to the selected problems for $F|no-wait|C_{\max}$ was also extracted from the literature. Due or specific deadlines for the test problems were extracted from the study by Garcia (2016). Then, we calculated tightness factor (henceforth TF) for specific dates, which is presented in the following table. We considered 4 TF settings for each test problem. Overall, there were 32 test problems for $F|no-wait, d_i|C_{\max}$ and 8 test problems for $F|no-wait|C_{\max}$. Test problems with due date constraints were called Car+DD. We set 300 seconds as the maximum solution time for each problem. The numerical data of our models are presented in Table . According to this table, the solution found in the CPU time limit is not feasible. In addition, A. Mixed-integer programming model dominates the other formulation. To understand the complexity of the sizes of the models, the examples of two modes and a type of sources are presented. The information about variables and constraints is provided in Table I. The findings indicate that the first model has more binary variables and requires fewer constraints. However, in the second model, there are fewer binary variables and more constraints that need to be included. Since the number of continuous variables is equal to na for both models, they are discarded.

A. CONCLUSION

This study attempted to solve a flow shop problem in which tasks were consecutively carried out and there was no elapsed or wait time between them. The optimality and efficiency of our models were assessed using the makespan criterion. This NP-Hard problem was solved twice. We solved the problem once using Mixed-Integer Programming Model and another time using a Constraint Programming Model.

Table I. Computational results

			MIP Model			CP Model		
Problem	**Size n*m**	**Specific Deadline or TF**	**Best Feasible OFV Found**	**Optimality Proved**	**CPU Time**	**OFV**	**Optimality Proved**	**CPU Time**
Car01+DD	**11*5**	**TF=1**	8,152	No	300	8,152	No	300
		TF=2	8,168	No	300	8,164	No	300
		TF=3	NFS	No	300	NFS	No	300
		TF=4	NFS	Yes	4	NFS	Yes	23
Car02+DD	**13*4**	**TF=1**	8,646	No	300	8,465	No	300
		TF=2	9,139	No	300	9,002	No	300
		TF=3	NFS	No	300	NFS	No	300
		TF=4	NFS	Yes	298	NFS	No	300
Car03+DD	**12*5**	**TF=1**	9,170	No	300	9,091	No	300
		TF=2	9,148	No	300	9,120	No	300
		TF=3	NFS	No	300	NFS	No	300
		TF=4	NFS	Yes	305	NFS	Yes	37
Car04+DD	**14*4**	**TF=1**	9,674	No	300	9,798	No	300
		TF=2	NFS	No	300	NFS	No	300
		TF=3	NFS	No	300	NFS	No	300
		TF=4	NFS	Yes	4	NFS	No	300
Car05+DD	**10*6**	**TF=1**	9,159	No	300	9,159	No	300
		TF=2	9,454	No	300	9,454	No	300
		TF=3	11,537	Yes	174	11,537	No	300
		TF=4	NFS	Yes	25	NFS	Yes	6
Car06+DD	**8*9**	**TF=1**	9,690	Yes	10	9,690	Yes	24
		TF=2	9,690	Yes	10	9,690	Yes	28
		TF=3	9,690	Yes	10	9,690	Yes	25
		TF=4	NFS	Yes	290	NFS	Yes	9
Car07+DD	**7*7**	**TF=1**	7,705	Yes	2	7,705	Yes	1
		TF=2	7,705	Yes	2	7,705	Yes	1
		TF=3	7,705	Yes	2	7,705	Yes	1
		TF=4	NFS	Yes	14	NFS	Yes	0
Car08+DD	**8*8**	**TF=1**	9,372	Yes	11	9,372	Yes	49
		TF=2	9,372	Yes	11	9,372	Yes	56
		TF=3	9,573	Yes	11	9,573	Yes	69
		TF=4	NFS	Yes	12	NFS	Yes	3

Analyses were carried out and the performances of our proposed models were assessed. To this purpose, the size and computational complexity of each model were determined. Comparing the sizes and complexity of the models showed that the first model had more variables and needed fewer constraints. In terms of computational complexity, the first model had a significantly better performance than the second model. In addition, the first model could solve more problems optimally in a shorter time than the second model. Computational results illustrated that finding a feasible solution to $F|no-wait,d_i|C_{\max}$ was difficult when there were tight due dates. Numerical results revealed that the mixed-integer programming model outperformed the other formulation.

Developing lower and upper bounds for $F|no-wait,d_i|C_{\max}$ using tight factors is a promising direction for the future research. However, finding a feasible solution to problems with tight specific dates is demanding. As a result, developing a problem-solving approach that can efficiently generate feasible solutions seems useful. Moreover, the mathematical models developed in this study incorporate a large number of *A* constraints. Therefore, future studies might develop models with fewer *A* constraints.

REFERENCES

Afshar-Nadjafi, B. (2014). A solution procedure for preemptive multi-mode project scheduling problem with mode changeability to resumption. *Applied Computing and Informatics*, DOI: 10.1016/j.aci.2014.02.003.

Beasley, J. E. (1990). OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research Society, 41(11)*, 1069-1072.

Błażewicz, J., Pesch, E., Sterna, M., & Werner, F. (2005). The two-machine flow-shop problem with weighted late work criterion and common due date. *European Journal of Operational Research, 165(2)*, 408-415.

Błażewicz, J., Pesch, E., Sterna, M., & Werner, F. (2008). Metaheuristic approaches for the two-machine flow-shop problem with weighted late work criterion and common due date. *Computers & Operations Research, 35(2)*, 574-599.

Brah, S. (1996). A comparative analysis of due date based job sequencing rules in a flow shop with multiple processors. *Production Planning & Control, 7(4)*, 362-373.

Carlier, J. (1978). Ordonnancements a contraintes disjonctives. *RAIRO Recherche Operationnelle, 12*, 333-351.

Dhingra, A. & Chandna, P. (2010). Hybrid genetic algorithm for SDST flow shop scheduling with due dates: a case study. *International Journal of Advanced Operations Management, 2(3)*, 141-161.

Ebrahimi, M., Fatemi Ghomi, S., & Karimi, B. (2013). Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. *Applied Mathematical Modelling*.

Garcia, C. (2016). Resource-constrained scheduling with hard due windows and rejection penalties. *Engineering optimization, 48(9)*, 1515-1528.

Gowrishankar, K., Rajendran, C., & Srinivasan, G. (2001). Flow shop scheduling algorithms for minimizing the completion time variance and the sum of squares of completion time deviations from a common due date. *European Journal of Operational Research, 132(3)*, 643-665.

Grabowski, J. & Pempera, J. (2000). Sequencing of jobs in some production systems. *European Journal of*

Operational Research, 125, 535-550.

Gupta, J. N., Lauff, V., & Werner, F. (2000). On the solution of 2-machine flow shop problems with a common due date. *Operations Research Proceedings 1999*, Springer.

Hall, N. & Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research, 44*, 510-525.

Hasanzadeh, A., Afshari, H., Kianfar, K., eFathi, M., & Jadid, A. O. (2009). A GRASP algorithm for the two-machine flow-shop problem with weighted late work criterion and common due date. *IEEE International Conference on Industrial Engineering and Engineering Management*.

Hunsucker, J. & Shah, J. (1992). Performance of Priority Rules in a Due Date Flow Shop. *Omega, 20(1)*, 73-89.

Javadi, B., Saidi-Mehrabad, M., Haji, A., Mahdavi, I., Jolai, F., & Mahdavi-Amiri, N. (2008). No-wait flow shop scheduling using fuzzy multi-objective linear programming. *Journal of the Franklin Institute, 345(5)*, 452-467.

Kaminsky, P. & Lee, Z. H. (2002). On-line algorithms for flow shop due date quotation. University of California, Berkeley (California, USA). http://www.ieor.berkeley.edu/~kaminsky/papers/ddq_flowshop.pdf.

Pan, C. H. (1997). A study of integer programming formulations for scheduling problems. *International Journal of Systems Science, 28(1)*, 33-41.

Pan, J. C. H., & Chen, J. S. (2005). Mixed binary integer programming formulations for the reentrant job shop scheduling problem. *Computers & Operations Research, 32(5)*, 1197-1212.

Panwalkar, S. & Koulamas, C. (2012). An O(n^2) algorithm for the variable common due date, minimal tardy jobs bicriteria two-machine flow shop problem with ordered machines. *European Journal of Operational Research, 221(1)*, 7-13.

Raaymakers, W. & Hoogeveen, J. (2000). Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing. *European Journal of Operational Research, 126*, 131-151.

Rajasekera, J., Murr, M., & So, K. (1991). A due-date assignment model for a flow shop with application in a lightguide cable shop. *Journal of Manufacturing Systems, 10(1)*, 1-7.

Ramezanian, R., Aryanezhad, M., & Heydari, M. (2010). A Mathematical Programming Model for Flow Shop Scheduling Problems for Considering Just in Time Production. *International Journal of Industrial Engineering, 21(2)*.

Samarghandi, H. & ElMekkawy, T. Y. (2012). A meta-heuristic approach for solving the no-wait flow-shop problem. *International Journal of Production Research, 50(24)*, 7313-7326.

Sarper, H. (1995). Minimizing the sum of absolute deviations about a common due date for the two-machine flow shop problem. *Applied Mathematical Modelling, 19(3)*, 153-161.

Selen, W. J., & Hott, D. D. (1986). A mixed-integer goal-programming formulation of the standard flow-shop scheduling problem. *Journal of the Operational Research Society*, 1121-1128.

Stafford, E. F. (1988). On the development of a mixed-integer linear programming model for the flowshop sequencing problem. *Journal of the Operational Research Society*, 1163-1174.

Tang, H. B., Ye, C. M., & Jiang, L. F. (2011). A New Hybrid Particle Swarm Optimization for Solving Flow Shop Scheduling Problem with Fuzzy Due Date. *Advanced Materials Research, 189*, 2746-2753.

Tari, F. G., & Olfat, L. (2013). Heuristic rules for tardiness problem in flow shop with intermediate due dates. *The International Journal of Advanced Manufacturing Technology*, 1-13.

Tseng, F. T., Stafford Jr., E. F., & Gupta, J. N. (2004). An empirical analysis of integer programming formulations for the permutation flowshop. *Omega, 32(4)*, 285-293.

Wagner, H. M. (1959). An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly, 6(2)*, 131-140.

Wismer, D. (1972). Solution of the flowshop scheduling with no intermediate queues. *Operations Research, 20,* 689-697.

Ziaee, M., & Sadjadi, S. J. (2007). Mixed binary integer programming formulations for the flow shop scheduling problems. A case study: ISD projects scheduling. *Applied Mathematics and Computation, 185(1)*, 218-228.