An Enumeration Algorithm for the No-Wait Flow Shop Problem with Due Date Constraints

Hamed Samarghandi*, Mehdi Behroozi**

**Department of Finance and Management Science, Edwards School of Business, University of Saskatchewan, Saskatoon, Saskatchewan, Canada, S7N 5A7*
(e-mail: samarghandi@edwards.usask.ca)

***Department of Industrial and Systems Engineering, University of Minnesota*
111 Church Street S.E., Minneapolis, MN 55455, United States
(e-mail: behro040@umn.edu)

Abstract: This paper develops an enumeration algorithm for the no-wait flow shop scheduling problem with due date constraints. In this problem, waiting time is not allowed between successive operations of jobs. Plus, each job is accompanied by a due date which is dealt with as a hard constraint. The considered performance criterion is makespan. The problem is strongly NP-hard. In this research, a new modelling approach is developed for the problem. This new modelling technique and the resulting observations are incorporated into a new exact algorithm to solve the problem to optimality. To investigate the performance of the algorithm, a number of test problems are solved and the results are reported. Computational results demonstrate that the developed algorithm is significantly faster than the mathematical models.

Keywords: Scheduling; No-Wait Flow Shop; Due Date Constraints; Enumeration Algorithm; Optimality

1. INTRODUCTION

In the no-wait flow shop problem, a special case of the classical flow shop problem, no waiting time is allowed between successive operations of jobs. In other words, once processing of a certain job is started, no interruption is permitted between operations of the job. In this paper, completion of each job is associated with a due date, i.e., jobs must be completed before their due dates. Due date side-constraints are among the most applicable constraints in scheduling and sequencing literature because real-world jobs are usually accompanied by a deadline for completion (Hunsucker and Shah 1992). It is assumed that all the jobs are ready at time zero and the considered performance measure is makespan. According to the three-field notation of the scheduling problems (Graham et al. 1979), the problem can be designated as $F \mid nwt, d_j \mid C_{\max}$. Samarghandi (2015) proves that $F \mid nwt, d_j \mid C_{\max}$ is NP-hard. Hall and Sriskandarajah (1996) provide a comprehensive review of the applications of the problem. The literature is rich with studies that develop heuristic or metaheuristic methods in order to deal with no-wait flow shop problems with or without due dates constraints. For the case of $F \mid nwt, d_j \mid \gamma$, due date constraints have been traditionally considered as soft constraints. In other words, violating due date constraints has been permitted with the objective function of minimizing a measure of the tardiness (e.g., number of tardy jobs or number of late days). Tardiness measures have frequently been combined with other performance measures such as makespan, total flow time, etc.; however, due date constraints have rarely been studied as hard constraints. This is mainly due to the fact that generating a feasible solution for the problem, or proving that a feasible solution does not exist, turns into a very challenging task, especially when due dates are not too loose or too tight. Since no-wait flow shop problem with due date constraints is strongly NP-hard, several algorithms have been devised to deal with the problem (Rajasekera et al. 1991, Hunsucker and Shah 1992, Sarper 1995, Brah 1996, Gupta et al. 2000, Gowrishankar et al. 2001, Kaminsky and Lee 2002, Błażewicz et al. 2005, Błażewicz et al. 2008, Hasanzadeh et al. 2009, Dhingra and Chandna 2010, Tang et al. 2011, Panwalkar and Koulamas 2012, Ebrahimi et al. 2014, Tari and Olfat 2014, Samarghandi 2015). All of these methods first relax the due date constraints and then solve the no-wait scheduling problem with a variant of lateness measure in the objective function by means of a metaheuristic or a heuristic algorithm. This paper introduces a new modelling approach for the no-wait flow shop problem and proves a number of theorems based on the characteristics of the $F \mid nwt, d_j \mid C_{\max}$. Afterward, an enumeration algorithm is proposed to solve $F \mid nwt, d_j \mid C_{\max}$ to optimality. Computational results reveal that the proposed algorithm is significantly faster than the competitive methods.

2. Problem Description

In the considered $F \mid nwt, d_j \mid C_{\max}$ it is assumed that: 1) all jobs follow the same predefined order of operations; 2) no pre-emption or interruption is allowed; 3) no job can be

processed by more than one machine at the same time, and no machine can process more than one operation at the same time; 4) all jobs must visit all machines, possibly with zero processing time on some of the machines; and 5) there should be no waiting time between consecutive operations of a job. The following notation is used throughout the rest of this paper:

m	Number of machines
n	Number of jobs
J_j	Job j
p_{ij}	Processing time of i th operation of J_j
c_{jk}	Contribution of J_k to the objective function when placed immediately after J_j
S_{ij}	Starting time of i th operation of J_j
F_j	Finish time of J_j
d_j	Due date of J_j

A solution of $F\mid nwt\mid C_{\max}$ can be described with a sequence $\pi=(\pi_1,\pi_2,...,\pi_n)$ of n jobs. It should be noted that $F\mid nwt\mid C_{\max}$ is a permutation scheduling, i.e. the sequence of the jobs on all machines is the same. Hence, the contribution of job k when placed immediately after job j (c_{jk}) is not dependent to the machines. The algorithm of Samarghandi (2015) can be employed with small modifications to calculate $c_{jk};\ j,k=1,2,...,n;k\neq j$. Note that $c_{jj}=0;\ j=1,2,...,n$. Once all the contributions are extracted by the mentioned algorithm, the contribution matrix C can be formed. This matrix is a $(n+1)\times n$ matrix that lists the contribution of each job to the makespan if placed after a certain job in the sequence.

$$C=[c_{jk};\ j=0,1,...,n;k=1,2,...,n]=\begin{bmatrix} c_{01} & \cdots & c_{0n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{bmatrix} \quad (1)$$

3. Search Graph

Figure 1 describes a search graph that represents the $F\mid nwt,d_j\mid C_{\max}$:

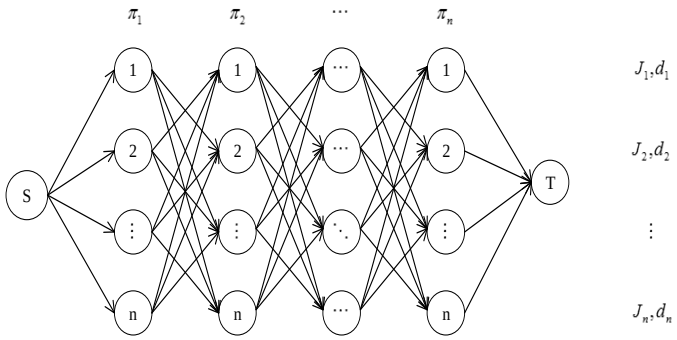

Figure 1 – The search graph respresenting $F\mid nwt,d_j\mid C_{\max}$

In this graph, the node which is located in the intersection of row $j;1\leq j\leq n$ and column $l;1\leq l\leq n$ represents job j if located in position l of permutation π; S and T are dummy jobs with zero processing times, which represent the start and the finish of the flow shop system. An arc exists between two nodes if and only if these nodes belong to two adjacent columns and they do not represent the same job.

A feasible solution of $F\mid nwt\mid C_{\max}$ starts with S and ends with T; it includes one and only one node in each row and in each column. As a result, Figure 2 characterizes the permutation $\pi=(2,1,3)$ and represents a feasible solution of $F\mid nwt\mid C_{\max}$ with three jobs.

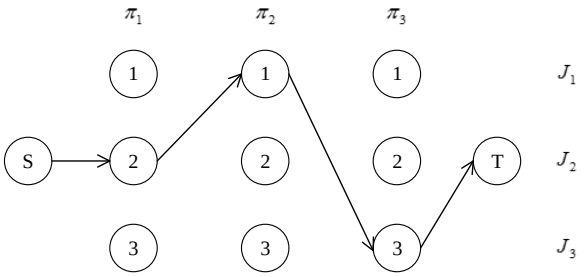

Figure 2 – A feasible solution of $F\mid nwt\mid C_{\max}$ with three jobs and three machines

Each arc $a_{jk};1\leq j,k\leq n$, when a_{jk} exists, can be labeled with c_{jk} as defined by (1). a_{Sj} represents the arc that connects S to J_j in column π_1 and is labeled with c_{0j}.

Observation 1: suppose that LP_{j,π_l} represents the longest path from S to the node in the intersection of column π_l and row j.

If $LP_{j,\pi_l}\leq d_j;\forall j\in\{1,2,...,n\},\forall l\in\{1,2,...,n\}$, then the due date constraints can be removed and the problem reduces to $F\mid nwt\mid C_{\max}$.

Observation 2: if $LP_{j,\pi_n} \le d_j; \forall j \in \{1,2,...,n\}$, then the due date constraints can be removed and the problem reduces to $F \mid nwt \mid C_{\max}$.

Observation 3: if $\exists j \in \{1,2,...,n\} \mid LP_{j,\pi_n} \le d_j$, then the due date constraints for J_j can be removed from the problem.

Observation 4: suppose that SP_{j,π_l} represents the shortest path from S to the node in the intersection of column π_l and row j.

If $\exists j \in \{1,2,...,n\} \mid SP_{j,\pi_l} > d_j, \forall l \in \{1,2,...,n\}$, then the problem is infeasible.

If $\exists j \in \{1,2,...,n\} \mid SP_{j,\pi_l} > d_j, \forall l \in \{1,2,...,n\}$ or $\exists l \in \{1,2,...,n\} \mid SP_{j,\pi_l} > d_j, \forall j \in \{1,2,...,n\}$, then the problem is infeasible.

4. The Enumeration Algorithm

The following algorithm represents the enumeration algorithm that solves $F \mid nwt, d_j \mid C_{\max}$ to optimality.

1. If $\exists j \in \{1,2,...,n\} \mid SP_{j,\pi_1} > d_j$, stop. The problem is infeasible.

2. If $LP_{j,\pi_n} \le d_j; \forall j \in \{1,2,...,n\}$, remove the due date constraints to reduce the problem to $F \mid nwt \mid C_{\max}$.

3. Calculate $SP_{j,\pi_l}; l \in \{2,3,...,n\}, j \in \{1,2,...,n\}$. If $\exists j \in \{1,2,...,n\} \mid SP_{j,\pi_l} > d_j; l \in \{2,3,...,n\}$, remove the corresponding node and all of its arcs from the graph G; call the remaining graph G'.

4. Find the shortest path between S and T with attention to the definition of the feasible solution of $F \mid nwt \mid C_{\max}$. If the found shortest path does not violate any of the due date constraints, it is optimal; compute the total contribution values of this path to calculate the makespan. Otherwise, proceed to step 5.

5. This step describes an enumeration sub-algorithm to solve G' to optimality. The objective of this sub-algorithm is to fathom all of the paths of the modified search graph (or G') from S to T until the optimum solution is found. The root node is S.

5.1. Branch from S to all of the nodes in π_1. Define l as the index for the positions in the permutation; in other words, l represents the current column in G'. Set $l \leftarrow 1$. Objective function value for node $j; j \in \{1,2,...,n\}$ is $C_j^l = c_{0j}$. Fathom all nodes in G' for $l > 1$.

5.2. Assume that $C_q^l = \max_j \left\{ C_j^l \mid j = 1,2,...,n \right\}$ such that j is not selected yet; update the current node to q; break the ties by random selection, unfathom all the nodes in column $t \mid t > l$, and branch from q to all of its adjacent nodes in G'; calculate $C_j^{l+1} \leftarrow C_q^l + c_{qj}$, $j \in \{1,2,...,n\}$ and q and j are adjacent.

5.3. Fathom the nodes that violate the due date of their respective jobs in column $l+1$, and go to step 5.6 if $l \ne n-1$; otherwise proceed to step 5.4. Note that if due date constraints are violated when $l = 1$, according to step 1 the problem is infeasible.

5.4. Compare $C_j^{l+1}; j \in \{1,2,...,n\}$ with $C_{\max}^{best}$, the makespan of the best-known feasible solution (if the list of the complete feasible solutions is not empty); if $C_j^{l+1} > C_{\max}^{best}; j \in \{1,2,...,n\}$, fathom node j in column $l+1$.

5.5. If $l = n-1$ and there is at least one node in column $l+1$ which is not fathomed yet, then the paths to such nodes define different feasible solutions each with makespan which is at least as desirable as $C_{\max}^{best}$. Accordingly, compare the makespan of such nodes with each other and update $C_{\max}^{best}$ with the best found makespan. Then, fathom all the nodes in column $l+1$ and proceed to 5.6.

5.6. If all of the nodes in $l+1$ are fathomed, then fathom the current node and proceed to 5.6.1. Otherwise, set $l \leftarrow l+1$ and go to step 5.2.

5.6.1. If there are nodes in the current column l, which have not yet been selected or fathomed during the course of the algorithm, do not change the value of l; go to step 5.2. Otherwise proceed to 5.6.2.

5.6.2. Set $l \leftarrow l-1$. If $l = 0$, stop. Report $C_{\max}^{best}$ and its corresponding route as the optimum solution. If the list of the feasible solutions is empty, the problem is infeasible. Otherwise, restart step 5.6 from the beginning. ■

Numerical results will be presented in the next section.

Table 1 –Computational Results

Problem	Size n*m	Due date Tightness Factor	Model of Samarghandi (2015) (T=600)	Enumeration Algorithm		
				T=60	T=300	T=600
Sam01+DD	7*7	TF=1	**7705, 2**	**7705, 0**	**7705, 0**	**7705, 0**
		TF=2	**7705, 2**	**7705, 0**	**7705, 0**	**7705, 0**
		TF=3	**7705, 2**	**7705, 0**	**7705, 0**	**7705, 0**
		TF=4	**NFS, 14**	**NFS, 0**	**NFS, 0**	**NFS, 0**
Sam02+DD	8*8	TF=1	**9372, 11**	**9372, 0**	**9372, 0**	**9372, 0**
		TF=2	**9372, 11**	**9372, 0**	**9372, 0**	**9372, 0**
		TF=3	**9573, 11**	**9573, 0**	**9573, 0**	**9573, 0**
		TF=4	**NFS, 12**	**NFS, 0**	**NFS, 0**	**NFS, 0**
Sam03+DD	8*9	TF=1	**9690, 10**	**9690, 0**	**9690, 0**	**9690, 0**
		TF=2	**9690, 10**	**9690, 0**	**9690, 0**	**9690, 0**
		TF=3	**9690, 10**	**9690, 0**	**9690, 0**	**9690, 0**
		TF=4	**NFS, 290**	**NFS, 0**	**NFS, 0**	**NFS, 0**
Sam04+DD	10*6	TF=1	**9159, 334**	**9159, 2**	**9159, 2**	**9159, 2**
		TF=2	**9454, 224**	**9454, 0**	**9454, 0**	**9454, 0**
		TF=3	**11537, 174**	**11537, 0**	**11537, 0**	**11537, 0**
		TF=4	**NFS, 25**	**NFS, 0**	**NFS, 0**	**NFS, 0**
Sam05+DD	11*5	TF=1	**8152, 3966**	**8152, 17**	**8152, 17**	**8152, 17**
		TF=2	**8164, 3402**	**8164, 9**	**8164, 9**	**8164, 9**
		TF=3	NFS	**NFS, 0**	**NFS, 0**	**NFS, 0**
		TF=4	**NFS, 4**	**NFS, 0**	**NFS, 0**	**NFS, 0**
Sam06+DD	12*5	TF=1	9084	**9084, 54**	**9084, 54**	**9084, 54**
		TF=2	9120	**9120, 25**	**9120, 25**	**9120, 25**
		TF=3	NFS	**NFS, 0**	**NFS, 0**	**NFS, 0**
		TF=4	**NFS, 305**	**NFS, 0**	**NFS, 0**	**NFS, 0**
Sam07+DD	13*4	TF=1	8465	9002	**8465, 226**	**8465, 226**
		TF=2	9002	**9002, 11**	**9002, 11**	**9002, 11**
		TF=3	NFS	**NFS, 0**	**NFS, 0**	**NFS, 0**
		TF=4	**NFS, 298**	**NFS, 0**	**NFS, 0**	**NFS, 0**
Sam08+DD	14*4	TF=1	9674	10613	9699	9699
		TF=2	NFS	**NFS, 24**	**NFS, 24**	**NFS, 24**
		TF=3	NFS	**NFS, 6**	**NFS, 6**	**NFS, 6**
		TF=4	**NFS, 4**	**NFS, 0**	**NFS, 0**	**NFS, 0**
Sam09+DD	15*6	TF=1	13472	15999	14991	14976
		TF=2	14666	15809	15014	14031
		TF=3	NFS	**NFS, 59**	**NFS, 59**	**NFS, 59**
		TF=4	**NFS, 3**	**NFS, 1**	**NFS, 1**	**NFS, 1**
Sam10+DD	16*7	TF=1	9017	9419	9419	9402
		TF=2	8977	9451	9432	9402
		TF=3	9262	NFS	9374	9374
		TF=4	NFS	**NFS, 11**	**NFS, 11**	**NFS, 11**
Sam11+DD	17*5	TF=1	11371	12680	12627	12625
		TF=2	NFS	NFS	NFS	NFS
		TF=3	NFS	NFS	**NFS, 137**	**NFS, 137**
		TF=4	**NFS, 2**	**NFS, 1**	**NFS, 1**	**NFS, 1**
Sam12+DD	18*9	TF=1	8904	10980	10886	10813
		TF=2	9232	11199	10943	10943
		TF=3	NFS	NFS	NFS	NFS
		TF=4	**NFS, 54**	**NFS, 2**	**NFS, 2**	**NFS, 2**
Sam13+DD	19*8	TF=1	17970	21204	21108	21023
		TF=2	NFS	NFS	NFS	NFS
		TF=3	NFS	NFS	NFS	NFS
		TF=4	NFS	**NFS, 2**	**NFS, 2**	**NFS, 2**
Sam14+DD	20*10	TF=1	31199	37045	36754	36754
		TF=2	34399	NFS	NFS	NFS
		TF=3	NFS	NFS	NFS	NFS
		TF=4	NFS	**NFS, 17**	**NFS, 17**	**NFS, 17**
Percent of efforts with optimum solution			**44.64%**	**66.07%**	**69.64%**	**69.64%**

5. Computational Experiments

The enumeration algorithm was coded by Microsoft Visual C++ 2013. All the numerical experiments were performed on a PC equipped with a 2GHz Intel Pentium IV CPU and 2 GB of RAM. To perform the computational analysis, a number of test problems generated by Samarghandi (2015) were selected. Numerical results of the enumeration algorithm were compared to the results of the developed mathematical model of Samarghandi (2015).

Best solutions of the enumeration algorithm for the test problems is reported at $T = 60$, $T = 300$ and $T = 600$ seconds; for the case of the mathematical model of Samarghandi (2015), the best solution is reported only for $T = 600$ seconds.

In the following tables, OFV represents objective function value and all of the CPU times are reported in seconds. The time when the optimal solution was found is reported as well. For instance, according to Table 1, the optimal solution of Sam01 with due date tightness factor 1 is 7705; this solution has been found by the mathematical model of Samarghandi (2015) after 2 seconds. Numbers in boldface indicate that the reported solution is optimal. Therefore, NFS in boldface means that the problem has no feasible solutions; however, non-bold NFS means that although the algorithm has not been able find a feasible solution in the given time, the problem may or may not have feasible solutions. Computational supremacy of the developed algorithm over the mathematical model of Samarghandi (2015) is evident from Table 1.

6. Conclusions

The no-wait flow shop problem with due date constraints and makespan criterion has been considered in this paper. The problem is strongly NP-hard. A graph modelling of the problem as well as an exact enumeration algorithm that employs this modelling have been presented based on the definition of the job contributions. Computational experiment has been conducted to compare the performance of the developed enumeration algorithm with mathematical models from the literature. Computational results illustrate that as the problem size grows, finding a feasible solution for $F \mid nwt, d_j \mid C_{\max}$ is not an easy task. Numerical results reveals that the enumeration algorithm outperforms the other formulations. Finally, developing tight lower and upper bounds for $F \mid nwt, d_j \mid C_{\max}$ is an interesting future research direction.

7. References

Błażewicz, J., E. Pesch, M. Sterna and F. Werner (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., E. Pesch, M. Sterna and F. Werner (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.

Dhingra, A. and P. Chandna (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., S. Fatemi Ghomi and B. Karimi (2014). "Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates." Applied Mathematical Modelling **38**(9-10): 2490-2504.

Gowrishankar, K., C. Rajendran and G. Srinivasan (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.

Graham, R. L., E. L. Lawler, J. K. Lenstra and A. R. Kan (1979). "Optimization and approximation in deterministic sequencing and scheduling: a survey." Annals of discrete mathematics **5**: 287-326.

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

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

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

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

Kaminsky, P. and Z.-H. Lee (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.

Panwalkar, S. and C. Koulamas (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.

Rajasekera, J., M. Murr and K. So (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.

Samarghandi, H. (2015). "A particle swarm optimisation for the no-wait flow shop problem with due date constraints." International Journal of Production Research **53**(9): 2853-2870.

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.

Tang, H. B., C. M. Ye and L. F. Jiang (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. and L. Olfat (2014). "Heuristic rules for tardiness problem in flow shop with intermediate due dates." The International Journal of Advanced Manufacturing Technology **71**(1-4): 381-393.