Flowshop Scheduling Problem - Introduction

The challenge has as focus the ***PERMUTATION FLOWSHOP PROBLEM***, in the ***MONO-OBJECTIVE*** case, the objective function to be minimized being the overall completion time for all the jobs – the ***MAKESPAN***. From a notational point of view, there are two ways of specifying a scheduling problem in a formal context, the initial notation being proposed by Conway et al. [CMM67], the second one, more widely used in the literature, being developed by Graham et al. [GLLK79], consisting of three distinct fields - $\alpha/\beta/\gamma$ – for describing the problem, as follows:

- α field indicates the structure of the problem;
- β field cumulates a set of explicit constraints (not implied by the internal semantic structure – for example, for flowshop, a job cannot start its execution on a machine if it's still under processing on the previous one);
- γ field indicates the objective(s) to be optimized.

A complete description of possible values for the above mentioned fields was proposed by T'kindt and Billaut [TB02], the enumeration of these values being out of scope for this paper, considering the targeted purposes. Also, a review of various scheduling problems currently researched in the literature was proposed by Lee [LLP97] and Pinedo [Pin95]. At this time, a multitude of scheduling problems are under research, rendering possible an abstract general presentation by grouping the main classical problems in five distinct classes:

- **workshops with only one machine - $\alpha_1 =$** : there is only one machine which must be used for scheduling the given jobs, under the specified constraints;

- **flowshop - $\alpha_1 =$ F**: there is more than one machine and each job must be processed on each of the machines – the number of operations for each job is equal with the number of machines, the j^{th} operation of each job being processed on machine j;

- **jobshop - $\alpha_1 =$ J**: the problem is formulated under the same terms as for the flowshop problem, having as specific difference the fact that each job has associated a processing order assigned for its operations.

- **openshop - $\alpha_1 =$ O**: the same similarity with the flowshop problem, the processing order for the operations being completely arbitrary - the order for processing a job's operations is not relevant; any ordering will do.

- **mixed workshop - $\alpha_1 =$ X**: there is a subset of jobs for which a fixed processing path is specified, the other jobs being scheduled in order to minimize the objective function.

Permutation Flowshop Scheduling Problem – Definition

The permutation flowshop represents a particular case of the flowshop scheduling problem, having as goal the deployment of an optimal schedule for N jobs on M machines.

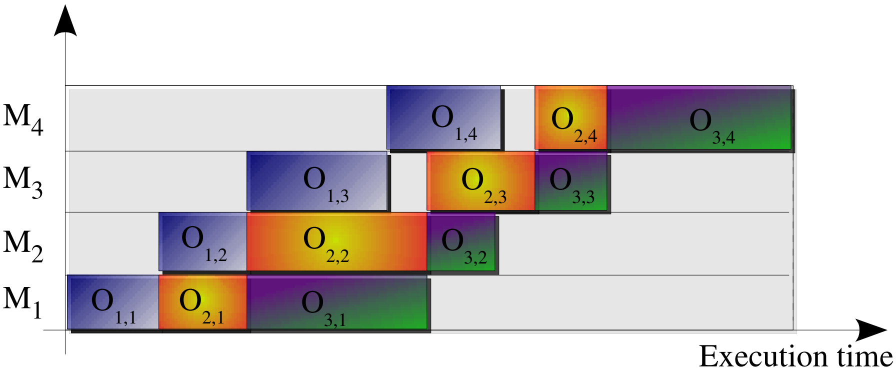
Solving the flowshop problem consists in scheduling *n* jobs ($i = 1...n$) on *m* machines ($j = 1...m$). A job consists in *m* operations and the j^{th} operation of each job must be processed on machine *j*. So, one job can start on machine *j* if it is completed on machine *j-1* and if machine *j* is free. Each operation has a known processing time $p_{i,j}$.

For the permutation flowshop the operating sequences of the jobs are the same on every machine. If one job is at the i^{th} position on machine 1, then this job will be at the i^{th} position on all the machines.

As a consequence, for the permutation flowshop problem, considering the makespan as objective function to be minimized, solving the problem means determining the permutation which gives the smallest makespan value. In the above specified context, a job J_i can be seen as a set of operations, having one operations for each of the M machines:

- $J_i = \{ O_{i1}, O_{i2}, O_{i3}, ..., O_{iM} \}$, where O_{ij} represents the j^{th} operation of J_i;
- O_{ij} operation must be processed on the M_j machine;
- for each operation O_{ij} there is an associated processing time p_{ij}.

Notationally, the problem is referenced by **F/permu/C_{max}**, considering as objective function to be minimized the overall processing time – the makespan. An example of a permutation flowshop problem schedule is shown in the below figure:



Let $\Pi_1, \Pi_2, \Pi_3, ..., \Pi_N$, be a permutation. Computing the finish time $C(\Pi_\iota, j)$ for the ith job of the given permutation Π and the machine j, can be done as follows:

$$\begin{array}{ll}
C(\Pi_1, 1) = p_{\Pi 1},1 & \\
C(\Pi_\iota, 1) = C(\Pi_{\iota-1}, 1) + p_{\Pi\iota},1, & i = 2, ..., N \\
C(\Pi_1, j) = C(\Pi_\iota, j - 1) + \ p_{\Pi 1},j, & j = 2, ..., M \\
C(\Pi_\iota, j) = \max\{C(\Pi_{\iota-1}, 1), C(\Pi_1, j - 1)\} + \ p_{\Pi\iota},j, & i = 2, ..., N; j = 2, ..., M
\end{array}$$

under these specifications, the value of the objective function, the makespan, C_{max}, is given by $C(\Pi_N, M)$ – completion time for the last operation on the last machine.

FSP Approaches

More complex approaches consider layered machine functional structuring, each layer being responsible for the processing of only one operation; according to problem type, the machines might be disposed in only one layer or in multiple layers. Also, another general class of complex problems considers scheduling under the terms of general assignment, each operation being dependent on a set of machines for completion.

Different criteria may be used for evaluating a schedule, the classical most used one being the overall completion time for all jobs on the given set of machines, known as makespan and notationally specified by $\mathbf{C_{max.}}$ Problem objectives are usually specified as an objective function to be minimized or as constraints that have to be satisfied for an instance to be considered a valid solution.

The challenge addresses flowshop problems – $F/\beta/\gamma$ - targeted at this time to makespan optimization permutation flowshop problems - all jobs must be scheduled in the same order for all the machines, this being an intrinsic constraint.

As a brief introduction for flowshop problems, a separation can be performed by considering the addressed number of machines: there are flowshops limited to only one or two machines – $F1/\beta/\gamma$ - as well as problems with a variable number of machines according to each instance specifications – $F2/\beta/\gamma$. All flowshop problems belonging to the second mentioned class are known to be strongly NP-hard. A review including a large number of different flowshop problem types may be found in [Esp98, TB02].

FSP Approaches – Exact Methods

Different ways of solving in exact manner the flowshop scheduling problem have been proposed over the time, considering also multi-criteria context definitions.

A B&B approach for flowshop problems with sequence dependent setup times has been developed by Rios-Mercado, Bard. Kohler [RMB99] as well as an B&C approach [RMB98], Steiglitz solving two machines flowshops by using different methods, exact and metaheuristic [KS75]. Also, Carlier & Neron proposed a solving method for hybrid flowshop problems [CN00].

Other directions, not addressed by the challenge[1], include multi-objective approaches, reusing techniques for mono-objective flowshop; such a method has been developed by Sayin and Karabati for a bi-objective FSP [SK99] - the ε-constraint method. Considering the fact that the flowshop problem is known to be strongly NP-hard, exact methods are restricted to small instances due to execution time; the importance of such methods being essential for future research, distributed approaches are developed for solving larger scale problem instances.

Benchmarks and Results

A very well known set of benchmarks has been made available by Taillard*, a subset of these benchmarks being presented in the following table – the last column indicates whether the given bound result is exact or just an upper bound approximation.

Benchmark	[Tai93a]	N	M	C_{max} bound	Ref	Exact
ta_20_5_01	tai001	20	5	1278	[Tai93a]	Yes
ta_20_10_01	tai011	20	10	1582	[Tai93a]	Yes
ta_20_20_01	tai021	20	20	2297	[Tai93a]	Yes
ta_50_5_01	tai031	50	5	2724	[Tai93a]	Yes
ta_50_10_01	tai041	50	10	2991	[Vae95]	Yes
ta_50_20_01	tai051	50	20	3850	[NS03]	No
ta_100_10_01	tai071	100	10	5770	[NS96]	Yes
ta_100_20_01	tai081	100	20	6202	[NS03]	No
ta_200_10_01	tai091	200	10	10862	[Vae95]	Yes
ta_500_20_01	tai111	500	20	26059	[Vae96]	No

*http://ina.eivd.ch/Collaborateurs/etd/problemes.dir/ordonnancement.dir/ordonnancement.html

[1] Please note that the challenge addresses ONLY the MONO-OBJECTIVE permutation flowshop, the objective function to be minimized being the MAKESPAN.

Bibliography

[CMM67] R. Conway, W. Maxwell and L. Miller. *Theory of scheduling*. Addison-Wesley, 1967.

[CN00] J. Carlier and E. Neron. An exact method for solving the multi-processor Flow-Shop. *RAIRO – Recherche Operationnelle*, 34(1) :1-25, 2000.

[Dav85] L. Davis. Job-Shop scheduling with genetic algorithms. In *International Conference on Genetic Algorithms and their Applications*, pages 136-140. Lawrence Erlbaum Associates, 1985.

[DEO03] A. Doye, O. Engin and C. Ozkan. A new artificial immune system approach to solve permutation flow-shop scheduling problem. In *Turkish Symposium on Artificial Intelligence and Neural Networks* – TAINN'03, volume 1, 2003.

[Esp98] M. L. Espinouse. *Flowshop et extensions: chevauchement des taches, indisponibilite des machines et systeme de transport*. PhD thesis, University of Joseph Fourier-Grenoble 1, December 1998.

[GLLK79] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A.H.G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Annals of Discrete Mathematics*, volume 5, pages 287-326. 1979.

[GP02] J. Grabowski and J. Pempera. New block properties for the permutation flowshop problem with application in tabu search. *Journal of the Operational Research Society*, 52(2): 210-220, 2002.

[HJ95] M.J.M. Heijligers and J.A.G. Jess. High-level synthesis scheduling and allocation using genetic algorithms based on constructive topological scheduling techniques. In *2nd IEEE International Conference on Evolutionary Computation ICEC'95*, pages 61-66, 1995.

[HL03] M. Haouari and T. Ladhari. A branch and bound based local search for the flowshop problem. *Journal of the Operational Research Society*, 54(10): 1076-1084, 2003.

[IYM03] H. Ishibuchi, T. Yoshida, and T. Murata. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transaction on Evolutionary Computation, 7(2): 204-223, 2003.*

[KS75] W.H. Kohler and K. Steiglitz. Exact, approximate, and guaranteed accuracy algorithms for the flow-shop problem n/2/F/F. *Journal ACM*, 22(1): 106-114, 1975.

[LLP97] C-Y. Lee, L. Lei, and M. Pinedo. Current trends in deterministic scheduling. *Annals of Operations Research*, 70: 1-41, 1997.

[NS96] E. Nowicki and C. Smutnicki. A fast tabu search algorithm for the flow shop problem. *European Journal of Operational Research*. 91: 160-175, 1996.

[Pin95] M. Pinedo. *Scheduling: Theory, Algorithms and Systems*. Prentice-Hall, 1995.

[PR98] S. Parthasarathy and C. Rajendran. Scheduling to minimize mean tardiness and weighted mean tardiness in flowshop and flowline-based manufacturing cell. *Computers and Industrial Engineering*, 34(2): 531-546, 1998.

[RMB98] R. Z. Rios-Mercado and J. F. Bard. Computational experience with a branch-and-cut algorithm for flowshop scheduling with setups. *Computers and Operations Research*, 25(5): 351-366, 1998.

[RMB99] R. Z. Rios-Mercado and J. F. Bard. A branch-and-bound algorithm for permutation flow shops with sequence-dependent setup times. *IIE Transactions*, 31(8): 721-731, 1999.

[SK99] S. Sayin and S. Karabati. A bicriteria approach to the two-machine flow shop scheduling problem. *European Journal of Operational Research*, (113): 435-449, 1999.

[Stu98a] T. Stutzle. An ant approach for the flow shop problem. In *Proceedings of the 6th European Congress on Intelligent & Soft Computing (EUFIT'98)*, volume 3, pages 1560-1564. Verlag-Mainz, 1998.

[Stu98b] T. Stutzle. Applying iterated local search to the permutation flow shop problem. Technical report, Darmstadt University of Technology, Computer Science Departement, April 1998.

[TB02] V. T'kindt and J.C. Billaut. *Multicriteria Scheduling – Theory, Moldels and Algorithms*. Springer-Verlag, 2002.

[Yam02] T. Yamada. A pruning pattern list approach to the permutation flowshop scheduling problem. *Essays and Surveys in Metaheuristics*. 12(1): 641-651, 2002.

[YL04] K-C, Ying and C-J. Liao. An ant colony system for permutation flow-shop sequencing. *Computers and Operational Research*, 31(5): 791-801, 2004.

[YS01] Z. Yong and N. Sannomiya. An improvement of genetic algorithms by search space reductions in solving large-scale flowshop problems. *EEJ Transactions on Electronics, Information and Systems*, 121-C(6): 1010-1016, 2001.