Giffler and Thompson algorithm for Flexible Job Shop Scheduling Problem

František Koblasa
Technical University of Liberec

Giffler and Thompson algorithm for Flexible Job Shop Scheduling Problem

František Koblasa [1]

[1] *Department of Manufacturing system, Faculty of Mechanical Engineering, Technical University of Liberec, Studentská 2, 461 17 Liberec, Czech Republic, frantisek.koblasa@tul.cz,*

ABSTRACT

Giffler and Thompson constructive algorithm (GT) is well known algorithm to construct schedule for theoretical Job Shop Scheduling Problem (JSP). However, practical scheduling problems are not that often classical JSP today. This article presents one of modified problems (model), which is already known, and presents simple modification to GT algorithm to solve Flexible Job Shop Scheduling Problem (FJSP). Algorithm is tested on mentioned theoretical FJSP using priority rules.

Keywords: Flexible Job Shop Scheduling Problem, Giffler and Thompson algorithm, Priority rules.

1 Introduction in to Flexible Job Shop Scheduling Problem

The necessity of scheduling grow up recent years due to rapidly changing customer order, consumer demand for variety, reduced product life cycles, changing markets with global competition and rapid development of new processes and technologies. These and several other conditions are defying mostly contradictory requirements for manufacturing e.g. minimizing inventory and maintain customer delivery due dates. It is possible satisfy these requirements thanks to efficient, effective and accurate scheduling plan. The Job-Shop Scheduling Problem (JSP) is one of the most popular scheduling models existing in practice [1].

1.1 Literature review

A classic Job Shop Scheduling Problem (JSP) has a set of *n* jobs processed by a set of *m* machines. Each job is processed on machines in a given order with a given processing time, and each machine can process only one type of operation. In the modern manufacturing plant, a machine has the capability of processing more than one type of operation [3], which is the problem that can not be solved by classical JSP model. This leads to a modified version of JSP called the Flexible JSP (FJSP).

Notwithstanding that today's companies are using multipurpose machines todays production planning based systems (ERP, APS, MES) are usually not able to solve neither optimize FJSP. In case that this kind of the system is able to work with the technological order variants, instead of keeping technological order for each of the variants, they usually simplify FJSP to JSP forcing user (planer, foreman) to choose technology variant.

There is also less literature concerning about FJSP. FJSP was first proposed by Brucker and Schlie [4]. They proposed a polynomial algorithm to solve the assignment and scheduling problems in FJSP with two jobs. However literature concerning about FJSP arise in past several years.

J. C. Chen [5] presents types of FJSP. For type I FJSP, jobs have alternative operation sequences and alternative identical or non-identical machines for each operation. The problem is to select the operation sequences for jobs and determine the job processing orders on machines. For type II FJSP, jobs can have only fixed operation sequences, but alternative identical or non-identical machines for each operation. The

Primary author contact:
František Koblasa, Department of Manufacturing system, Faculty of Mechanical Engineering, Technical University of Liberec, Studentská 2, 461 17 Liberec, Czech Republic, frantisek.koblasa@tul.cz

problem is to arrange jobs to machines according to their operation sequences. This article focuses on the second type of FJSP.

Haoxun Chen et al. [6] present generalized approach to genetic algorithm solution of FJSP type II. F. Pezzella et al. presents an evolutionary algorithm which use priority rules for job sequencing [7]. Jie Gao et al. hybridize genetic algorithm using random walk search algorithm [8] to solve FJSP with parallel variable neighborhood. Same subject is solving M. Yazdani et al. [9] by several neighborhood structures. Mohammad Saidi-Mehrabad et al. Use tabu search for JSP [10]. Decent architecture to solve FJSP made Nhu Binh Ho,et. Al [11]. Practical example of glass factory was published by R. Alvarez-Valdes et al. [12]

There is wide variety of methods (genetic, hybrid, tabu-search algorithms) presented but very few of them are presenting general idea to solve FJSP together with constraints that are usual in daily practice. There were presented several modification and supportive algorithm to Giffler and Thompson algorithm to take in account constraints as setup and pass setup [13], shift work systems [14] and influence between these modifications [15]. To solve „multi-purpose machine“ constrain i.e. FJSP GT algorithm will be modified as in the previous work on projected scheduling application.

2 Modified Giffler and Thompson algorithm for FJSP

Projected scheduling application works with the active and non-delay schedules. In active schedules (A) the processing sequence is such that no operation can be started any earlier without delaying some other operation or violating technological constraints. In a non-delay schedule no machine is kept idle when it could start processing some operation. These schedules form a sub-class of the active [16].

In the algorithms we schedule operations one at a time. We can say that one operation is schedulable if all the operations which must precede it within its job have already been scheduled. General idea is to take in account all variants of schedulable operation in conflict set (step 4).

Algorithm will proceed by so many stages as many operations require technology order.

At stage *t* is

P_t – partial schedule of (t-1) scheduled operations
S_t – set of operations schedulable at stage t, i.e., all the operations that must precede those in S_t are in P_t
V_t – set of variant operations, i.e all variants of schedulable operations
e_t – the earliest time that operation o_k in S_t could be started
f_t – the earliest time that operation o_k in S_t could be finished, that is $f_t = o_k + p_k$, where p_k is the processing time of o_k
(Note: statements for non-delay schedule are in brackets)

1. Let $t = 1$ with P_t being null. S_t will be the set of all operations with no predecessors, those that are first in their job.
2. Generate V_t by selecting each operations and its variant based on St. So V_t will be set of all variant operations of S_t including (together with different machines, process times etc.)
3. Find $f_t^* = \min_{o_k \text{ in } V_t}\{f_t\}$ (ND - $e_t^* = \min_{o_k \text{ in } V_t}\{e_t\}$) and machine M^* on which f_t^* occurs. If there is a choice for M^* choose arbitrarily.
4. If there is the operation o_k in V_t which
   a) Requires M^*
   b) $e_t < f_t^*$ (ND - $e_t^* = e_t$)
   Choose o_k from S_t
5. Move to next stage by
   I. Adding o_k to P_t, to create P_{t+1}
   II. Deleting o_k from S_t and creating S_{t+1} by adding to S_t the operations that directly follows o_j in its job (unless o_k completes its job). Delete V_t

Primary author contact:
František Koblasa, Department of Manufacturing system, Faculty of Mechanical Engineering, Technical University of Liberec, Studentská 2, 461 17 Liberec, Czech Republic, frantisek.koblasa@tul.cz

III. Increment *t* by 1

6. If there are any operations left unscheduled, go to 2. Otherwise, stop.

3 Testing example

Presented idea is tested on basic problem (model). This model contains 3 jobs and 6 machines [17]. These machines are able to handle all operations, with different process times to complete (Table 1).

Table 1: Example benchmark 3 jobs - 6 machines: processing time and ordering operation.

		M(0)	M(1)	M(2)	M(3)	M(4)	M(5)
J(0)	O(0)	10	7	6	13	5	1
	O(1)	4	5	8	12	7	11
	O(2)	9	5	6	12	6	17
	O(3)	7	8	4	10	15	3
J(1)	O(0)	15	12	8	6	10	9
	O(1)	9	5	7	13	4	7
	O(2)	14	13	14	20	8	17
J(2)	O(0)	7	16	5	11	17	9
	O(1)	9	16	8	11	6	3
	O(2)	6	14	8	18	21	14

The most used objective funcion in theoretical scheduling models is difference between the start and finish of a sequence of jobs i.e. makespan.

3.1 Constructing schedules

It was used 5 priority rules to construct active (A) and non delay (ND) schedules. These rules are most common in daily practices thus it is not expected that any of these rules are going to reach optimal makespan, which is for presented problem (model) 18 time units.

Priority rules and their application aspect:

- SPT (shorter processing time) to finish small orders first;
- MWKR (most work remaining) for the case that we whish not to focus on one job;
- FCFS (first come first served) to make a process of continuous flow;
- Random – knowing that we have in the conflict set (using Active schedule generation) all "active operations" and we have no information that can support our decision to choose a rule. Random is the best in that instant.

Most difficult for scheduling application is MWKR, because there are no clear expectations on job routing policy on machines. Following method is used to apply this rule:

- It is summed all shortest processing times to construct overall job most work remaining criteria i.e.

$$MWKR\ (J_0, J_1, J_2) = (13, 18, 14) \qquad (1)$$

- When job is assigned to machine shorter processing time of all variant jobs is removed. i.e. if $J_0\,(O_0,M_2)$= 6 proc time is scheduled as first operation most work remaining will be due

Primary author contact:
František Koblasa, Department of Manufacturing system, Faculty of Mechanical Engineering, Technical University of Liberec, Studentská 2, 461 17 Liberec, Czech Republic, frantisek.koblasa@tul.cz

$J_0\,(O_0,M_5)$= 1:

$$\text{MWKR}\,(J_0, J_1, J_2) = (12, 18, 14) \qquad (2)$$

3.2 Results

There were tested 5 priority rules as mentioned above. There were generated 10 random schedules. Table 2 shows results for active (A) and non delay (ND) schedules, where Random rule average from 1é results is marked (10) and best schedule form random schedule (b). There is also comparison of the result and its percentage distance from optimum (Opt.dis. %)

Table 2: Computation results

	A	Opt dis.%	ND	Opt dis.%
SPT	18	0,00%	37	105,56%
MWKR	30	66,67%	46	155,56%
FCFS	27	50,00%	30	66,67%
Random (b)	24	33,33%	30	66,67%
Random (10)	26	44,44%	36,4	102,22%

As table 2 shows thanks to SPT rule in the scheme of active schedules it was obtained optimum for this problem (model). It shoves that even priority rule can be pretty useful and there is possibility to get optimum by simple priority rule as shorter processing time (see figure 1). The worst rule during testing on this specific problem is MWKR. Random rule which is usually underestimated priority rule shows that its average distance from the optimum is not that great and best result from 10 schedules is second best from all rules in this particular problem.

Primary author contact:
František Koblasa, Department of Manufacturing system, Faculty of Mechanical Engineering, Technical University of Liberec, Studentská 2, 461 17 Liberec, Czech Republic, frantisek.koblasa@tul.cz

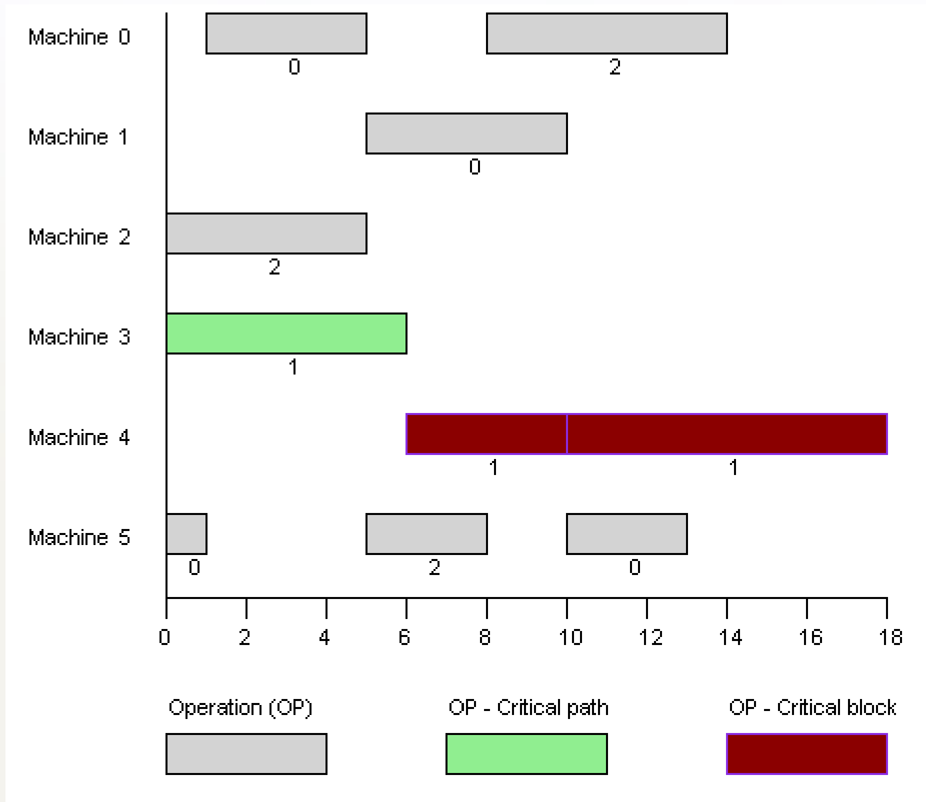

Figure 1: Optimal schedule

4 Conclusion

Giffler and Thompson algorithm is the most used algorithm to construct schedules for JSP. This article shows that with particular modification can be used for FJSP. Result shows, that priority rules as easiest and generally fastest way to construct schedule can give us particularly efficient results.

Further work will focus on implementing constraints as setups, shift work, transport. Than work will focus on developing local search scheme based on Single swap local search and on genetic algorithm based on random keys representation. The goal will be to develop optimizations algorithm that will be able to optimize both FSP and JFSP.

References

[1] JAIN A.S., MEERAN S., Deterministic job-shop scheduling: Past, present and future, *In: European Journal of Operation Research* 113 (2) (1998) 390–434.

[2] CHAN FTS, WONG TC, CHAN LY: Flexible job-shop scheduling problem under resource constraints. In: *Int J Prod Res* (2006) 44(11):2071–2089

[3] BRUCKER P. : Scheduling algorithms, 2nd edition, Springer, Heidelberg 1998

[4] BRUCKER P, SCHLIE R: Job-shop scheduling with multi-purpose machines. Computing 45(4):369–375 (1990)

[5] CHEN J.C., CHEN K. H , WU J. J. , CHEN C. W. : A study of the flexible job shop scheduling problem with parallel machines and reentrant process. *In: The International Journal of Advanced Manufacturing Technology (2008).* ISSN 0268-3768.

[6] CHEN H., IHLOW J. LEHMANN C.: A Genetic Algorithm for Flexible Job-Shop Scheduling *In:Proceedings of the IEEE International Conference on Robotics & Automation Detroit, Michigan May 1999*

Primary author contact:
František Koblasa, Department of Manufacturing system, Faculty of Mechanical Engineering, Technical University of Liberec, Studentská 2, 461 17 Liberec, Czech Republic, frantisek.koblasa@tul.cz

[7] PEZZELLAA F., MORGANTIA G., CIASCHETTI G.: A genetic algorithm for the Flexible Job-shop Scheduling Problem. *In: Computers & Operations Research 35* (2008) 3202 – 3212 ISBN: 978-0-7695-3311-7

[8] GAOA J., SUNA L., GENB M.: A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problem. *In:Computers and Operations Research* Volume 35 , Issue 9 (September 2008) Pages 2892-2907 ISSN:0305-0548

[9] YAZDANI M. ZANDIEH M. AMIRI M.: Flexible job-shop scheduling with parallel variable neighborhood search algorithm *In:Expert Systems with Applications: An International Journal* a Volume 37 , Issue 1 (accepted in print January 2010) Pages 678-687 ISSN:0957-4174

[10] MEHRABAD M. S. FATTAHI P.: Flexible job shop scheduling with tabu search algorithms *In: The International Journal of Advanced Manufacturing Technology* PublisherSpringer London 2007 ISSN0268-3768

[11] HO, NHU BINH & TAY, JOC CING & LAI, EDMUND M.-K: An effective architecture for learning and evolving flexible job-shop schedules,*In: European Journal of Operational Research, Elsevier*, vol. 179(2), pages 316-333, June 2007. ISSN: 0377-2217

[12] ALVAREZ-VALDES R., FUERTES A., TAMARIT J.M., GIMENEZ G. ,RAMOS R.: A heuristic to schedule flexible job-shop in a glass factory. *In: European journal of operational research* (2004) ISSN 0377-2217

[13] KOBLASA, F.: Modified Giffler and Thompson algorithm for setup and pass-setup constrain. *In: 3. ročník mezinárodní konference „Výrobní systémy dnes a zítra 2008“.* Sborník anotací příspěvků, Liberec 27. - 28. 12. 2008. Liberec: TU v Liberci - KVS, 2008. ISBN 978-80-7372-416-0.

[14] KOBLASA F.,MANLIG F.: Job shop scheduling problem with multi-shift work system. *In:"Vědecká pojednání" XV / 2009, ACC Journal*, č. publikace: 55-050-09, čj. 50/09, Technická univerzita v Liberci, ISSN 1803-9782,

[15] KOBLASA F.: Constrained scheduling - shifts and setups. *In: Recenzovaný sborník abstraktů z Mezinárodní Baťovy konference pro doktorandy a mladé vědecké pracovníky 2009.* Univerzita Tomáše Bati ve Zlíně. 2009.ISBN 978-80-7318-811-5

[16] French, S.: Sequencing and Scheduling: An Introduction to the Mathematics of the Job-shop.In:*Ellis Horwood limited*, West Sussex, England 1982, ISBN 0-470-272

[17] LIOUANE, N., I. SAAD, S. HAMMADI and P. BORNE, Ant Systems & Local Search Optimization for Flexible Job-Shop Scheduling Production, *In: International Journal of Computers, Communications & Control*, Vol. II, No. 2, pp. 174-184, 2007.

Reviewer

Doc. Dr. Ing. František Manlig , Department of manufacturing systems, Faculty of mechanical engineering, Technical university of Liberec, Studentská 2, 461 17 Liberec 1

Primary author contact:
František Koblasa, Department of Manufacturing system, Faculty of Mechanical Engineering, Technical University of Liberec, Studentská 2, 461 17 Liberec, Czech Republic, frantisek.koblasa@tul.cz