JOB SHOP SCHEDULING PROBLEM WITH MULTI-SHIFT WORK SYSTEM

F. Koblasa, F. Manlig

Technická univerzita v Liberci
Fakulta strojní, Katedra výrobních systémů
Studentská 2, 461 17 Liberec, Česká republika
frantisek.koblasa@tul.cz
frantisek.manlig@tul.cz

Abstract

The job shop scheduling problem is a well known NP-hard optimization problem. There are a lot of optimization techniques to optimize theoretical problems. It is necessary to use additional constraints to optimize real world cases.

This paper presents job shop scheduling in the multi-shift work system environment. Firstly, there is presented a modification of Giffler and Thompson (GT) Constructive Algorithm (CA). Then the above mentioned modification is tested on a well known theoretical problem by CA, Local Search (LS) and Genetic Algorithm (GA). The contribution also focuses on the time span of these methods.

1. Introduction

Nowadays, companies are forced to provide their customers with a large variety of products in the low volume production, with a view to different requirements from product consumers and low stock levels. This kind of manufacture is called the job shop. The most frequent algorithm for job shop scheduling used in a theoretical way is the Giffler and Thompson [1] scheduling algorithm for an active schedule scheme and the modified non-delay schedule [2]. To be able to use this algorithm for real world cases, it is necessary to include several constraints as a setup [3], and also work shifts.

Shift work, which is typical mainly for manufacturing organizations, has a clear impact on the function of a company. We can increase production up to 300%, compared to the day shift system, by using the 3-shift system. It has also an impact on utilization of manufacturing resources (machines). Another reason is represented by (typically in automotive, chemical, metallurgic and textile industry) expenses bounded with stopping and restarting the manufacture.

A very demanding scheduling is a significant disadvantage of the work shift system, besides high requirements on laborers, especially in the night and morning shifts. A typical shift work system is the 3-shift system with 8 resp. 7.5-hour shifts. This system is unusually used in two modifications – the morning shift begins at noon (0:00) or at 22:00. The shift work setting is also depending on the company culture, which is usually taken up from foreign countries resp. “mother companies”, or is given by the optimized process.

2. The Constructive Algorithm for the multi-shift work system

The above mentioned Giffler and Thompson algorithm for the active (non-delay) schedule is based on the scheduling operation one by one. It is searching through available operations for further alternatives with the earliest ending (starting) time. These times are usually influenced by constraints as setup times [4].

In the matter of single setup times the constructing solution is very easy. The only thing that we do is simply sum up the setup time with the processing time to get the overall time that the job will occupy. So we need only additional information about the setup time length respecting the job and machine, same as in the process time.

Pass-setup is possible when we can make setup without completing the preceding job (Figure 1). This is usually possible thanks to the setup tools, jigs, standard parts etc. The goal is to make an account of the unused time on the machine minimizing makespan, total flow time, total weighted flow time, etc. The key thing considering the Giffler and Thompson algorithm is to set a correct starting time of both the setup and process. For this we need to gather information about the available machine starting and ending times of the preceding job [5].

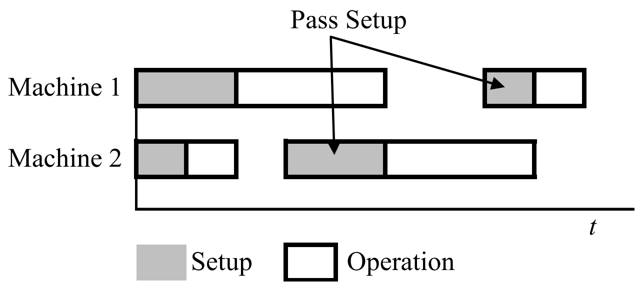

Figure 1: Gantt chart with pass-setup

The starting and ending time of the operation is also influenced by the shift work system. To test the shift work modification we use the classical job shop problem FT10 [6], which was already modified – for every job there has to be a pass setup with 50 time units. The shift work modification has to take into account the additional information about the machines as:

- The shift system
- The number of shifts in a day
- The duration of shifts
- The duration of breaks
- The beginning of the first shift

The FT10 model is then modified as shown in table 1

Machine	N. of avaiable shifts	Shift duration [min]	Break duration [min]	Shift system	Begining of 1st shift
0	1	720	30	2	22:00
1	2	720	30	2	22:00
2	1	480	30	3	22:00
3	2	480	30	3	22:00
4	3	480	30	3	22:00
5	1	480	30	3	22:00
6	2	480	30	3	22:00
7	3	480	30	3	22:00
8	1	480	30	3	22:00
9	2	480	30	3	22:00

Table 1 Shift work modification

Shift work algorithm (figure 2) consists of three main parts. The first part contains the procedure to get the relatively earliest staring time of both the job (operation) and the earliest available time on the machine. The absolute times are updated simultaneously. These times have to respect (they can not start in) both breaks and the idle shift (the shift where the machine is idle). The algorithm is right shifting - looking for the first further feasible date of these times.

The second part has to modify the relative starting time, which is influenced by the shifts, breaks, relative starting time, the earliest starting time on the machine and the setup time. This model includes the before mentioned pass-setup. So the earliest start of the setup is when the machine is available. The earliest time of setup end could be at the end time of the precedence job (the earliest starting time of the job). These times are already modified in the first part of the algorithm. The algorithm is now left shifting (finding an earlier possible start of the operation) the starting time of the setup, controlling length of the setup time and the earliest starting time on the machine. The algorithm simply shifts the start of the operation till the earliest starting time on the machine is reached or the setup time is “depleted”. The setup time is reduced by the available time, ignoring the idle shifts and breaks. The overall starting time of the setup is set.

The setup time and process time of the operation is iteratively added to the starting time to obtain the end time of the setup in the third part of the algorithm. The procedure first checks if the relative starting time extended by the setup is longer than the available time given by the shift system. In the case that it is longer, the end of the setup is prolonged by the time that remains to the beginning of the next shift break and by the idle shifts that follow (respecting the shift work system). The setup is reduced only by the remaining time to the beginning of the next shift.

If the relative starting time prolonged by the setup is not longer than the available time during the shift, the setup is simply added to the relative starting time to get the ending time of the setup.

The procedure of adding the processing time is the same as for the setup time using the end of the setup as the starting time of the operation.

The algorithm is generally working with these main types of data:

- e_t – absolute starting time of operation (e.g. 22:30 1.1.2009)
- e_{tr} – relative starting time of operation in selected time units (e.g. 30 [min])
- f_t – absolute ending time of operation
- f_{tr} – relative ending time of operation
- o_k – selected operation
- p_k – processing time of o_k
- se_k – setup time of o_k
- m_k – machine required by o_k
- m_t – absolute earliest possible starting time on machine which is required by o_k
- m_{tr} – relative earliest possible starting time on machine which is required by o_k
- sf_t – shift duration
- br_t – break duration
- br_{te} – duration of summed breaks and idle shifts, which are upcoming; depends on shift system
- sft_{tn} – relative number of selected time shift (e.g. 1^{st} – used for e_{tr} , m_{tr})
- $sft_m\{\}$ – available shift on machine

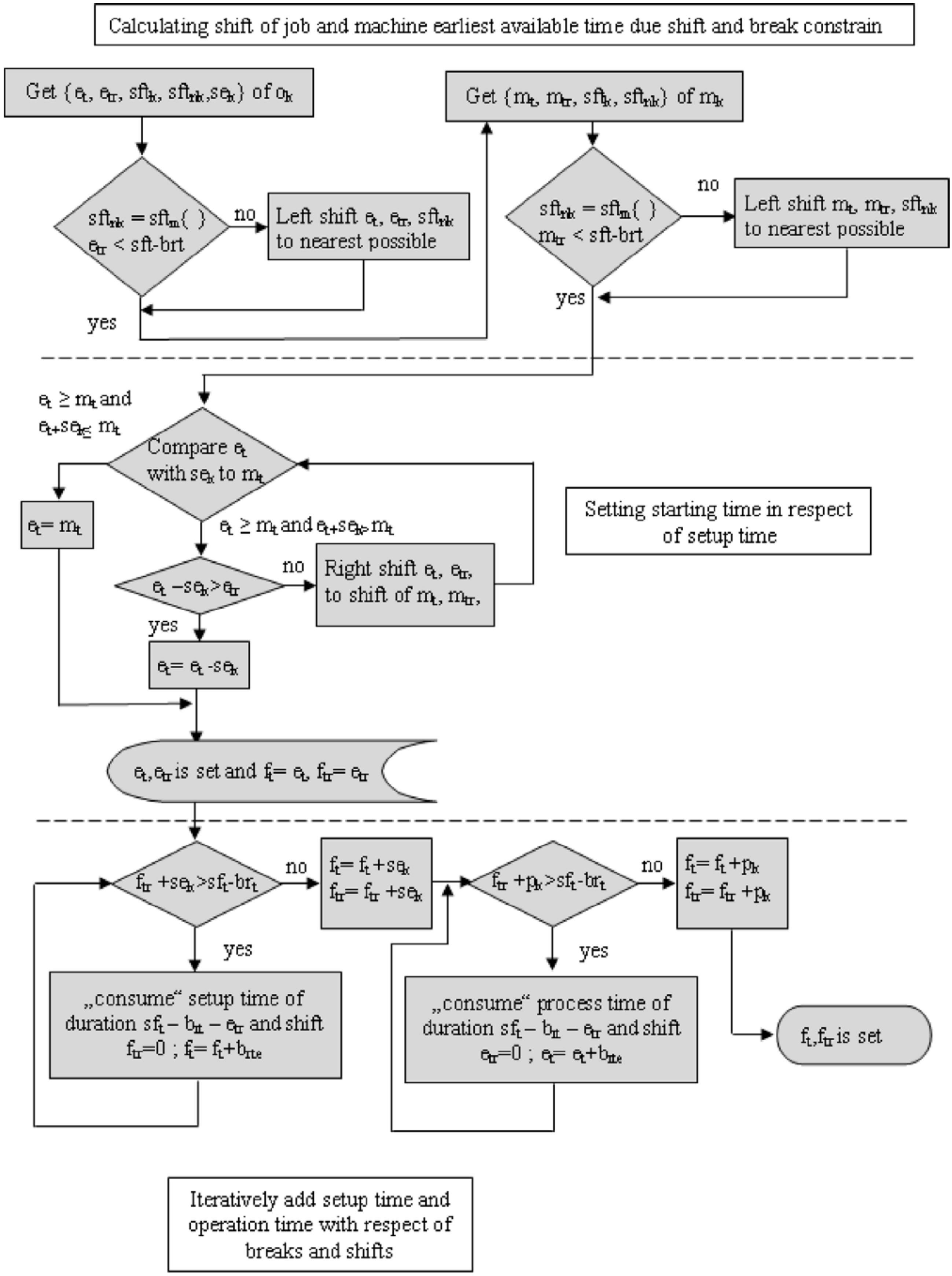

Figure 2: General scheme of shift work algorithm

3. Testing efficiency of the shift work scheme

We have chosen the modified FT10 theoretical problem, as described before, to test efficiency and effectiveness of the shift work scheme used in the GT algorithm. Every modification (constraint) as setup, pass setup and shift work influences not only effectiveness (reaching optimum solution) of the following optimization methods, but also efficiency (time span required for constructing schedule) of the constructive algorithm. We use constructive algorithm, Local search and Genetic algorithm with these settings:

- General
  - Schemes:
    - Active schedule scheme (A)
    - Non delay schedule scheme (ND)

- Constructive algorithm (CA)
  - Priority rules [7]:
    - Shorter processing time (SPT)
    - Most work remaining (MWKR)
- Local search (LS)
  - Ending condition: There is no better solution than previous in the new neighborhood.
  - Neighborhood : Single swap [8]
  - Initial solution : Given from CA using SPT and MWKR
- Genetic algorithm (GA) with chromosome correction
  - Population size: 100
  - Generations: 100
  - Parent selection : Roulette wheel
  - Crossover: Job based crossover (JOX)
  - Crossover coeff.: 0.6
  - Challenge strategy: Elite
  - Mutation: Clone control [9]

The test includes the measuring of the time span required to construct the schedule and optimization to compare results with (table 2) and without (table 3) shift work scheme. The optimality of the schedule is compared thanks to the time (time units) of completion of all jobs – makespan.

Without shift work				
Method	Rule	Active s.	Non delay s.	Time span
CA	SPT	1840	1855	1 s <
	MWKR	1937	1867	1 s <
LS	SPT	1827	1828	2 s <
	MWKR	1907	1863	2 s <
GA	-	1537	1564	620 s

Table 2 Makespan and time span comparison of CA, LS and GA without using shift work scheme.

With shift work				
Method	Rule	Active s.	Non delay s.	Time span
CA	SPT	4700	4744	1 s <
	MWKR	4541	4609	1 s <
LS	SPT	4591	4744	2 s <
	MWKR	4541	4609	2 s <
GA	-	3404	3440	660 s

Table 3 Makespan and time span comparison of CA, LS and GA with using shift work scheme

Comparing results we can assume several interesting things. We can see that using priority rules, we will get different results (in the view which is better). SPT rule using the active schedule scheme gives us the best result when we don't use shift work, but in the same model considering this constraint is one of the worse.

A local search with a single swap scheme was sufficient to optimize the schedule in the case of the model without any shift work constraint; but in the case of using this constraint an initial solution was too strong local optimum for this scheme.

The genetic algorithm was successful in both models. GA improves makespan in comparison with the best results from CA and LS significantly. Considering the scheme without the shift work, GA reduces makespan by 19% in an active schedule scheme and 16% in non delay schedule scheme. Considering shifts GA improves makespan approximately by 25% in both schedule schemes.

The time span of optimization is one of the most important criteria in the real world cases. As results showed, Shift work algorithm increases the time span by 6%. This increase is less than we expected, but we shall consider that testing the model was relatively small-scale and in the case of bigger problems can be more significant.

4. Conclusion

Inclusion of constraints as the setup and the shifts is very important to get an accurate schedule of the real process in the companies. Knowing that every other constraint than the technological order will increase the computation time of the schedule construct and following optimization, we must carefully analyze which, constraint is important and which not. Result show that the shift work constraint has had a great impact not only in the case of a greater makespan but also on the used priority rule, knowing that today's ERP/APS systems are used usually as only a one priority rule for scheduling.

Further investigation will focus on transportation constraint, together with testing of this constraint on modified theoretical and real case model.

Literature:

[1] Giffler B., Thompson G.: Algorithms for Solving Production Scheduling Problems. European Journal of Operational Research, 1960, vol. 8, pp. 487-503

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

[3] Artigues C., Lopez P., Ayache P. D.: Schedule Generation Schemes for the Job-Shop Problem with Sequence-Dependent Setup Times: Dominance Properties and Computational Analysis Annals of Operations Research, Volume 138, Number 1, September 2005 , pp. 21-52(32), Springer

[4] Vinod V., Sridharan R.: Scheduling a dynamic job shop production system with sequence-dependent setups: An experimental study Source, Robotics and Computer-Integrated Manufacturing archive, Volume 24 , Issue 3 (June 2008) Pages 435-449 ISSN:0736-5845

[5] Koblasa F.: Modified Giffler and Thompson algorithm for setup and pass-setup constrain. Proceedings of 3rd international conference "Výrobní systémy dnes a zítra 2008". Liberec 27-28.11 November 2008. ISBN 978-80-7372-416-0.

[6] Fisher H., Thompson G.L.: Probabilistic learning combinations of local job-shop scheduling rules", J.F. Muth, G.L. Thompson (eds.), Industrial Scheduling, Prentice Hall, Englewood Cliffs, New Jersey, 225-251., 1963

[7] Panwalkar S.S., Iskander W.: A survey of scheduling rules. Operations Research, Vol. 25, No.1, 45-60, 1977

[8] Koblasa F., Dias L.S., Oliveira J.A., Pereira G. : Heuristic Approach as a way to Improve Scheduling in ERP/APS Systems. Proceedings of 15th European Concurrent Engineering Conference (ECEC2008). Eds. A. Brito and J.M. Teixeira, 47-51, Porto April 2008. EUROSIS-ETI Publication. ISBN 978-9077381-399-7. (All EUROSIS Proceedings are ISI-Thomson and INSPEC referenced)

[9] Koblasa F., Dias L.S., Oliveira J.A.: Scheduling Optimization Using Local Search and Genetic Algorithm. "Vědecká pojednání" XIV / 2008, ACC Journal, ISBN 978-80-7372-379-8, ISSN 1801-1128, č. publikace: 55-083-08, čj. RE 106/08, Technická univerzita v Liberci

Doručeno redakci: 14. 4. 2009
Recenzováno: 16. 6. 2009
Schváleno k publikování: 23. 6. 2009

REIHENFOLGEPLANUNG IM MEHRSCHICHTBETRIEB

Das Reihenfolgeproblem in der Ablaufplanung ist bekanntes NP (nonpolynomialy) schwer Optimierungsproblem. Heutzutage benutzt man zur Lösung von theoretischen Problemen viele Optimierungsmethoden. In der Betriebspraxis ist es aber nötig, weitere Begrenzungen einzuführen, die meist in theoretischen Modellen nicht oder nur wenig berücksichtigt werden.
Dieser Beitrag präsentiert eine Lösung für einen Mehrschichtbetrieb. Zuerst wird hier eine Modifikation des Konstruktionsalgorithmus (CA) Giffler und Thompson beschrieben. Weiterhin werden die Ergebnisse der Testung dieses Algorithmus auf modifiziertem theoretischem Beispiel diskutiert. Die Aufmerksamkeit gilt dabei auch den Zeitansprüchen der Optimierung.

SZEREGOWANIE ZADAŃ W SYSTEMACH PRACY WIELOZMIANOWEJ

Szeregowanie zadań jest znanym trudnym problemem klasy NP (non-polynomial) w zakresie optymalizacji. Obecnie istnieje wiele technik wykorzystywanych do optymalizacji stosowanej w rozwiązywaniu problemów teoretycznych. Jednak w celu optymalizacji problemów rzeczywistych istnieje konieczność wprowadzenia dodatkowych ograniczeń, które w przypadku problemów teoretycznych nie są brane pod uwagę.
W artykule poruszono problem szeregowania zadań w środowisku pracy wielozmianowej. W pierwszej kolejności zaprezentowano modyfikację algorytmu konstruktywnego (CA) Giffler and Thompson (GT). Następnie algorytm ten testowano na zmodyfikowanym problemie teoretycznym wraz z optymalizacją heurystyczną, jaką jest lokalne przeszukiwanie (LS) oraz algorytm genetyczny (GA). Artykuł porusza również problem czasochłonności procesu optymalizacji.

ROZVRHOVÁNÍ V RŮZNORODÉM SMĚNNÉM SYSTÉMU

Rozvrhovací problém je známým NP- těžkým optimalizačním problémem. V dnešní době je známo mnoho optimalizačních technik pro řešení teoretických problémů. Pro optimalizaci reálných problémů je však třeba zavést další omezení, jež nejsou v teoretických problémech uvažována.
Tento článek prezentuje řešení rozvrhování v prostředí, kde jsou uplatňovány různé směnnostní systémy najednou. Článek nejdříve prezentuje modifikaci konstruktivního algoritmu (CA) Giffler a Thompson (GT) pro tento případ. Dále je tento algoritmus testován na modifikovaném teoretickém problému spolu s heuristickou optimalizací jako je lokální prohledávání (LS) a genetický algoritmus (GA). Článek také poukazuje na problém časové náročnosti optimalizace.