

JOB SHOP SCHEDULING IN MECHANICAL ENGINEERING

KOPECEK, P[avel]

Abstract: Common ERP systems do not solve the support for workshop management. This paper shows why and under what conditions computer support for workshop schedules is necessary. Demands on the quality of operation plans, factors having an influence on the scheduling process and criteria for selection of appropriate operations are discussed. A verification program was developed for testing the proposed hypotheses. This program proved that a repeated random selection and simulation of the production process could be a good tool for a small number of machines and production tasks. There is no need for sophisticated scheduling for a great number of machines and tasks, operational plans are enough for this purpose.

Key words: scheduling, production process, ERP, direct control of production, discrete optimization

1. INTRODUCTION

A detailed/precise/daily scheduling is made at the workshop level. If there are only a few scheduled items and workplaces and due times are relatively long, then a need for accurate schedules is more important than for a large number of short operations (the paradox of filling a rucksack with boxes – Fig. 1).

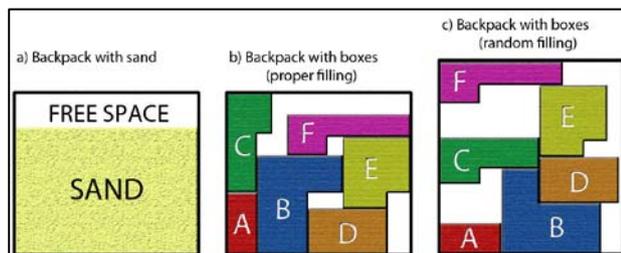


Fig. 1. Backpack with sand and boxes

An important property which is omitted in many theses is the sliding and open nature of an operation plan. A nice example of a sliding plan can be shown as a parable with a pipeline (Fig. 2).

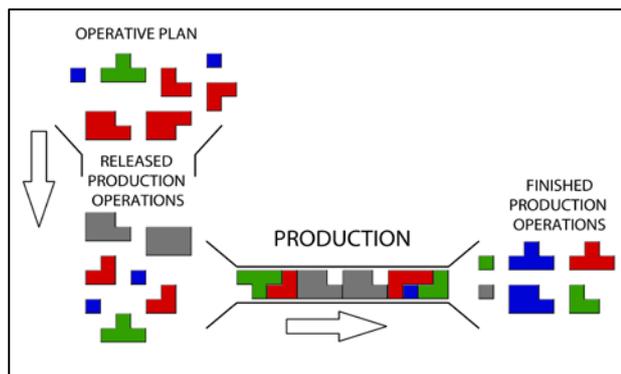


Fig. 2. Daily plan as a way through a pipeline

It does not matter how fast all tasks of an operative plan are fulfilled, but how to process operations without delays against due times and without idle times and overtimes at static balanced workplaces. Operations are released from operation plans in batches (once a day, once a week, and once in ten days).

New operations are released in batches from the operation plan. They are sequenced (pushed) by the detailed schedule into the production (pipeline) so as to avoid idle times (bubbles in the pipes). The production is a continuous process in time.

2. THEORETICAL SOLUTIONS

Whenever a task is specified, it is possible to precisely formulate it and try to find a method to solve it. If an analytical solution cannot be found, it is possible to try to find an iterative solution or a suitable simulation method which leads to the goal. If this fails, then some heuristic solutions could be found. In this case we usually stop trying to achieve the optimum solution, but look for solutions that are not bad and are as close to the optimum as possible.

The first option is to check all possibilities. Unfortunately, scheduling algorithms are very complex and known as ‘NP-complete’.

It is possible to solve a few problems algorithmically. All others can be divided into two groups:

Where a detailed schedule does not make sense, because a large number of scheduled tasks and workplaces leads to a convergence of static and dynamic usage of workplaces (queues are so long that it does not matter what we choose);

Where a detailed schedule is the point, then a heuristic solution could be proposed that is certainly better than the worst, in a better case better than average, and at best a little worse than the optimum.

There are special cases, for example the Johnson algorithm (Stevenson 2009) for two workplaces. Most books about operations management omit tasks of precise scheduling and leave these problems to be solved by workshop managers. Some authors have proposed modern approaches (Ridzoň 2002) – Genetic algorithms, (Ghoniem 2002, Knyttl 2010) – Shifting Bottleneck) with good results. In many cases they use makespan criteria and do not respect the sliding nature of operative plans. Also input parameters (capacities, overtimes, etc) are considered to be constant even though they may be changed by creative workshop managers.

It is also possible to use a brute force solution; this means checking all possibilities. Because $N!$ solutions exist; this enormous number of solutions could be reduced to a number which can be computed in a minute, let us say 10 000. Random permutations could be checked and the best one chosen.

However, there is no sure guarantee that a good solution can be found. If it is assumed that there is a one in a thousand chance of finding a relatively good solution, there is a 999 in a thousand chance of not finding a good solution. For 10 random selections there is still a 990 in a thousand chance of finding a bad solution, for 100 random selections 904 in a thousand, for 1

000 only 36 % and for 10 000 random selections only 0.005 % possibility of a bad solution.

3. HYPOTHESIS

The more operations and workplaces that are scheduled, the less is the need for a detailed schedule (example with the rucksack). For many large production workshops, an operative plan is adequate for workshop control purposes;

A random selection of permutations could be a good solution for a small number of workplaces and operations (say 10 workplaces and 20 tasks with 100 operations);

4. VERIFICATION PROGRAM

The main idea for the verification program is that if a set of well-balanced production data has been prepared then the program can also find a schedule which is good, better or a little worse than the original input data. It can be assumed that layouts of machines and transport times are ignored. Transport times are considerably shorter than processing and waiting times.

The goal of the verification program is to prove that using simulation and a random selection of operations in comparison with a deterministic method can lead to acceptable results.

The program is fully controlled by input data. As an example, a solution for 10 batches of parts, 40 operations and 6 various workplaces with 10 machines (one workplace can be represented with more fully reciprocal changeable machines) is presented.

The algorithm is based on simulation; in the simulated time the machine which will be free first is found. A suitable active operation (selected random, with shortest, longest operation time or the nearest due date) is selected for this machine, and the end of the operation is estimated in the simulated time, simultaneously the next operation in the routing of the batch becomes active in the simulated time. This is repeated until all operations are scheduled.

The criterion for the schedule quality is the sum of (linear and quadratic) delays of operation due dates (tardiness).

5. EXPERIMENT RESULT

The criterion of shortest (value of this criterion is 29) and longest (value 42) operation times gives according to expectation or worse results than according operation due dates (value 17).

For 50,000 random schedules, criteria results lie in the range 0 – 121, zero was achieved in 8 cases and values less than 15 (better than in the case of due dates) in 94 cases. Frequencies of criteria values are displayed linearly and logarithmically in Fig 3.

Also for 1,000 random experiments, the results were much better than when using deterministic criteria. Concurrently, it is possible to see that one random selection has a high probability of a very bad schedule.

More than 100 experiments with various data and numbers of random selections were performed. It could be assumed that this simple algorithm based on random selection and a relatively small number of repetitions could be a good way of solving practical tasks with a small number of machines and tasks.

The simple heuristic solution (random selection) is very fast (10 machines, 6 workplaces, 10 batches, 40 operations and 1000 repeated schedules needs 25 seconds on Pentium 4, 2.27 GHz, 1GB RAM).

(Knyttl2010) states that heuristics give much better results than scheduling rules. Experiments having been made with the developed validation program are in agreement with this contention.

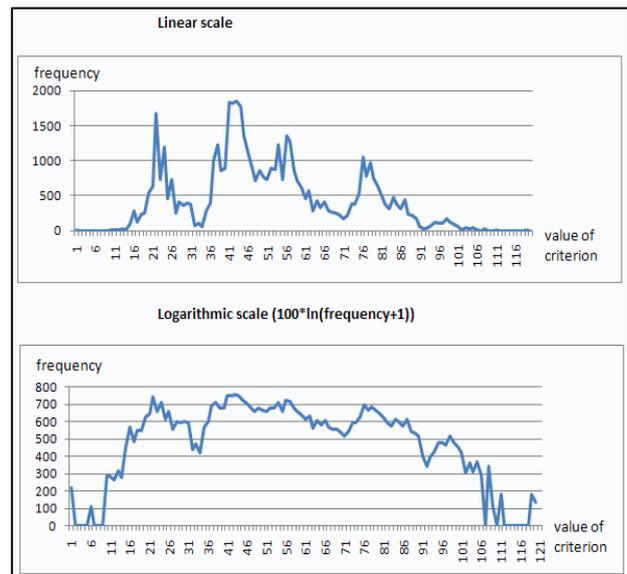


Fig. 3. Results of scheduling

6. CONCLUSION

Although the first successful attempts at a dynamic schedule were made in the 1970s, modern enterprise information systems have only been introduced en masse in the last ten years. When combined with advanced programming technology, this creates the preconditions for the successful development of tools for operational management of production. The subject of further research should be to find selection criteria, including the impact of significant factors, modification and verification of the scheduling algorithm to minimize delays to scheduled due dates.

Previous results support the hypothesis that a small number of operations in relation to the number of machines are needed to optimize the schedule.

It was experimentally demonstrated that there is a better solution than the selection of scheduled operations according to the operation due date. This solution can be found heuristically.

The research will be continued, the verification program will be tested on real production data with the goal to develop a simple scheduling software tool for an educational purpose.

7. ACKNOWLEDGEMENTS

This article was prepared with support from the Internal Science Foundation of the University of West Bohemia SGS-2010-065.

8. REFERENCES

- Ghoniem, A., S. *Static and Dynamic Job-Shop Scheduling Using Rolling-Horizon Approaches and Shifting Bottleneck Procedure*, Research Report, Polytechnic Institute and State University, Nantes, France, 2002
- Knyttl, J. *Dynamic Scheduling of Production* (in Czech language Dynamické rozvrhování výroby), Diploma Thesis, University of West Bohemia, 2010
- Ridzoň, M. *Program for Solution of Production Scheduling Using Evolution Algorithms* (in Slovak language Program pre riešenie plánovania výroby s použitím evolučných algoritmov), Diploma Thesis, Technical University of Košice, 2002
- Stevenson, W, J. *Operations management*, McGraw-Hill, 2009, ISBN 978-0-07-337784-1