ANALYSIS OF VARIABLE CHANGEOVER TIMES IMPACT ON THE REVENUE IN MANUFACTURING PROCESS

Jardzioch, A.; andrzej.jardzioch@zut.edu.pl
Skobiej, B.; bartosz.skobiej@zut.edu.pl

Abstract: *This paper presents a comparison of two manufacturing models performing prototype and small batch production. The first model considers variable changeover times which depend on a current tooling code of a machine and a tooling code required by a job. In the second model, changeover times are not taken into consideration. A non-classical evolutionary approach based on a genetic algorithm is proposed to schedule production jobs in both models. On the basis of the data obtained from the manufacturing company, 10 and 100 jobs were analyzed. The analysis made it possible to determine the impact of changeover times and various scheduling methods (Genetic Algorithm, Descending Delay Penalty, Earliest Due Date, Time Reserve, Shortest Processing Time, Longest Processing Time) on potential revenue.*

Key words: *changeover times, prototype production, small batch production, revenue, job scheduling*

1. INTRODUCTION

For many years, job scheduling has been an object of research. It was before the Second World War that the basis for contemporary scheduling Decision Support Systems (DSS) was laid in the United States (Gantt, 1919). Despite the passage of time, the issue of job sequencing is still the subject of current interest, and is present in numerous publications – not only those concerning production engineering (Gawiejnowicz, 2008; Chakraborty (Eds.), 2009; Kister et al., 2006). It is noteworthy that the process of job sorting has a great impact on the process of schedule creation, and is thus considered to be the basic optimization tool in production management (Lawler et al., 1993). Continuous improvement that has become a management strategy of contemporary enterprises, increasing computer performance and the development of such disciplines as computer optimization or artificial intelligence methods, results in regular attempts to solve well-known problems or to develop new, more realistic manufacturing models (Karger et al., 2009).

2. PROBLEM DESCRIPTION

The main issue investigated in this paper is the development of software supporting the management of job queues. The result of the work of a sorting algorithm is a schedule from which the arrangement of transportation routes in the production process is derived. The algorithm also determines the overall production time, delays, changeover times and related financial aspects. The focus of the present paper is on the variation in changeover times and delays in the production process that result from the use of different sorting algorithms, and its influence on revenue. The manufacturing subsystem analyzed consists of one machine that can be treated as a subsystem of identical or similar parallel machines that are grouped together, or as a group of machines in a process line. While the most frequently used sorting methods aim at minimizing production time, the focus of this study is on a different queuing criterion – revenue.

2.1 Revenue

In order to evaluate individual sortings, a criterion function was introduced. The function, adapted from (Wang et al., 2013), is dependent on the revenue from individual jobs reduced by a delay penalty. The function is described by the formula *(1)*.

$$R_k = \sum_{i=1}^{n} [\, r_i - (W_i * \max\{0, C_i - d_i\})\,] \tag{1}$$

where:
R_k – revenue from jobs in the k^{th} sorting,
i – job number,
n – number of jobs,
r_i – revenue from the i^{th} job,
W_i – weight penalty factor for the i^{th} job,
C_i – real time of the i^{th} job accomplishment,
d_i – due date of the i^{th} job.

Whenever negative numbers were obtained by calculating the difference C_i - d_i, i. e. when a job is accomplished in due time, the penalty factor W_i will not influence the revenue, since it is multiplied by 0. In unfavorable situations, the penalty may exceed potential revenue and generate loss. Despite the loss in case of one task, one cannot arbitrarily establish that the whole sorting is ineffective when compared to other ones.

2.2 Changeover times

In order to make the study more suitable for existing situations, times necessary to prepare a machine for a new job were also taken into consideration. These times are very important in case of prototype and small batch production. In simulation models, one very often assumes that machine changeover time is constant or that it is negligible. The times are variable, in fact, and depend not only do they depend on the type of a job to be processed, but also on what a given machine was processing before. In this paper, a method for defining variable changeover times necessary to prepare a machine for processing of a job was proposed.

For each job a tooling code was defined; for a machine a current tooling code was defined. If the current tooling is the same as the tooling required by a job, changeover time equals 0. If the code of the required tooling differs from the current tooling code of a machine, a changeover is necessary. The time needed for a changeover depends on the scope of necessary activities, and is included in the machine changeover times table.

In the present study, changeover times depend on the order of jobs under consideration and are added to processing times. The codes assigned to machines and jobs are integers ranging from 0 to the value set by the user.

2.3 Computational complexity

One has to bear in mind that optimization space is regarded to be discrete, and its exploration cannot be based on exhaustive search. While the exhaustive search for 3 jobs equals 3! possibilities (on the basis of the formula for permutations without repetition), the analysis of sorting for 100 jobs involve the exploration of optimization space consisting of about $93*10^{187}$ potential solutions. Additionally, it is important to note that the results of the present study are intended to support an enterprise that expects effective results for the number of jobs ranging from 100 to 300.

3. SCHEDULING ALGORITHMS

Six sorting algorithms were used in the present study.

3.1 Genetic Algorithm (GA)

In the non-classical genetic algorithm proposed, integer permutation encoding of individuals was used (Yu et. al. 1997). Each individual is a vector of job numbers to be accomplished, e.g. (1,2,3,4,5) for five jobs. Each job has the following attributes:

- job number i,
- potential revenue r_i,
- weight penalty factor W_i,
- processing time t_i,
- due date d_i,
- required tooling code k_i.

The first step is to randomly generate the first population of individuals. Then, the fitness of individual sortings is evaluated using the formula *(1)*. On the basis of the fitness vector obtained, the fittest – elite – individual is selected. The next step is the rank selection of the population. In consequence, a given percentage of the fittest individuals (excluding the elite) is retained for further evolution. The remaining individuals are removed from the population and replaced with random ones. With regard to crossover operator, examples of crossovers for permutation encoding are described in literature, e.g. PMX - Partially Matched Crossover (Goldberg, 1985) or SXX - Sub-tour Exchange Crossover (Yamamura et al., 1992). In the present study, however, the focus is on the analysis of the mutation operator, regarded to be the secondary genetic operator. Crossover operator was not taken into consideration.

In case of mutation operator, heuristic mutation was applied (Cheng & Gen, 1997). The idea of the mutation is shown in fig. 1. The last stage of genetic algorithm cycle is checking stopping criteria. In the present paper, two stopping criteria were specified: the number of iterations defined by the user and the computer memory utilization.

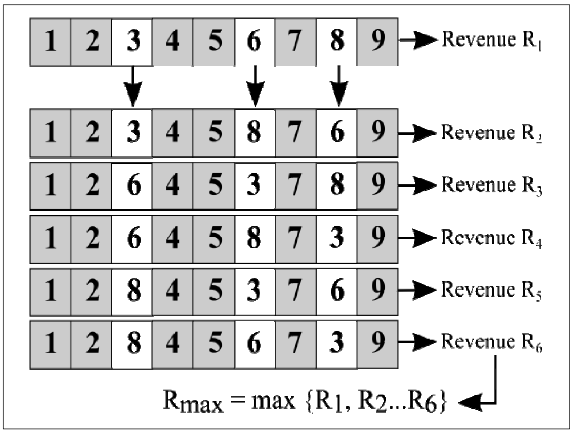

Fig.1. An example of 3-gene heuristic mutation for nine jobs

3.2 Descending Delay Penalty (DDP)

The DDP algorithm sorts jobs according to the weight penalty factor W_i, from the highest to the lowest value. In consequence, the jobs that may bear the greatest influence on revenue if delayed, are processed first.

3.3 Earliest Due Date (EDD)

The EDD algorithm sorts jobs according to due date, from the earliest to the latest one. It is one of the frequently used methods allowing for minimization of delays.

It is noteworthy that in case of job list of the same priority (identical penalty weight), it is the optimal algorithm for a one machine system with regard to the sum of tardiness.

3.4 Time Reserve (TR)

The Time Reserve algorithm sorts jobs according to increasing time reserve which is the difference of due date and processing time at t_0 point.

3.5 Shortest Processing Time (SPT)

The SPT algorithm sorts jobs according to the processing time, from the shortest to the longest one.

3.6 Longest Processing Time (LPT)

The LPT algorithm sorts jobs according to the processing time, from the longest to the shortest one.

4. RESEARCH

The study involved the analysis of two job lists. There were 10 jobs on the first list, and 100 jobs on the second one. Each list was sorted with the algorithms discussed in section 3. In addition, each list was examined with three different settings of changeover times: variable changeover times, constant changeover times equal 30, and lack of changeovers. The results included: revenue, overall production time, changeover times and the sum of tardiness.

5. RESULTS

Analyzing fig. 2, one can observe how the inclusion of changeover times influences financial effectiveness of sorting algorithms. While in case of the lack of changeovers 5 out of 6 sortings brought revenue, the inclusion of changeover times resulted in five algorithms generating lists of jobs making a loss. The only solution generating a job list bringing revenue was the genetic algorithm.

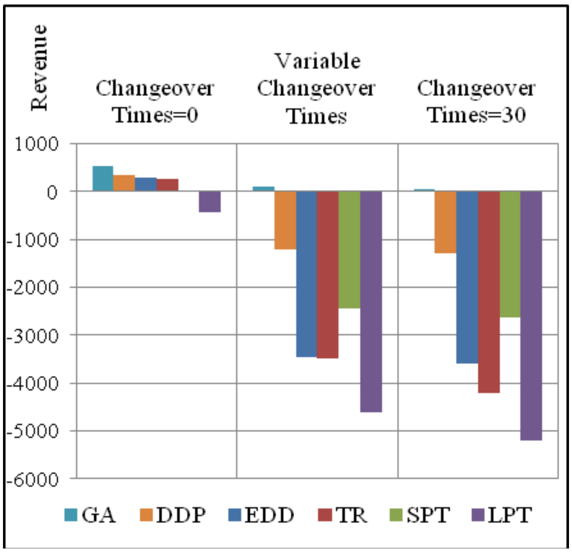

Fig.2. Changeover times impact on revenue for 10 exemplary jobs and 6 sorting algorithms

The influence of delays on revenue constituted an interesting interrelation that was observed in the course of the study. What follows from fig. 3, is that the comparable financial results achieved by DDP and SPT algorithms are not directly related to delays generated by these algorithms. This is because the revenue is influenced not only by tardiness, but – above all – the weight of delay penalty.

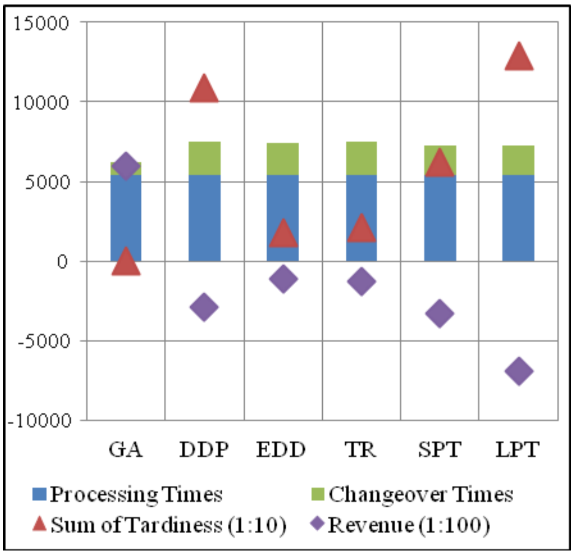

Fig.3. Revenue, tardiness and production time for 100 jobs with variable changeover times

Both in case of 10 and 100 jobs, the genetic algorithm is the only algorithm that allows for finding a sorting that brings revenue.

6. CONCLUSION

The results of the study show that the most frequently used classical sorting algorithms EDD, TR and SPT prove inefficient with regard to systems featuring changeover times. Since the times affect job delays, they also affect revenue. It can also be observed that the inclusion of penalties for delay in individual jobs significantly affects revenue. The results obtained with genetic algorithm turned out to be better than those generated by the remaining algorithms. Even the use of the DDP algorithm that minimizes the delay of jobs with the highest penalty weight did not result in a proper sequence. This is because even the slightest delay in job processing can often be the cause of high penalties.
Prioritizing jobs follows from the Pareto principle that says that 40% of customers generate 60% of revenue. Our goal is to identify those customers, prioritize their jobs and plan the production process accordingly.

7. REFERENCES

► Chakraborty, U. K. (Eds.) (2009). *Computational Intelligence in Flow Shop and Job Shop Scheduling*, Springer Science + Business Media, ISBN 9783642028366

► Chen B. et al. (1996). A New Heuristic for Three-Machine Flow Shop Scheduling, *Operations Research*, vol. 44, No. 6, ISSN 0030364X (Print), 15265463 (Online)

► Cheng R. & Gen M. (1997). Resource-constrained project scheduling problem using genetic algorithms, *Intelligent Automation and Soft Computing 3*, pp 273-286, ISSN 10798587 (Print), 2326005X (Online)

► Gantt, H. L. (1919). *Organizing for Work*, Harcourt, Brace, and Howe, New York. Reprinted by Hive Publishing Company, Easton, Maryland, 1973, ISBN 0879600462

► Gawiejnowicz, S. (2008). *Time-Dependent Scheduling*. Springer Science + Business Media, ISBN 9783540694465

► Goldberg, D.E. (1985). Alleles Loci and the TSP, *Proceedings of the First International Conference on Genetic Algorithms and Their Application*, Lawrence Erlbaum, New Jersey, pp 154-159, ISBN 0805804269

► Karger, D. et al. (2009). Scheduling Algorithms, *Algorithms and Theory of Computation Handbook*, Chapman & Hall/CRC Publishing, ISBN 9781584888185

► Kister, T.C. et al. (2006). *Maintenance Planning and Scheduling*, Elsevier Science Publishers B.V., ISBN 9780750678322

► Lawler, E.L. et al. (1993). Sequencing and Scheduling: Algorithms and Complexity. *Handbooks in Operations Research and Management Science*, Vol. 4., Logistics of Production and Inventory, pp445-522, Elsevier Science Publishers B.V., ISSN 0927-0507

► Wang, X. et al. (2013). A modified artificial bee colony algorithm for order acceptance in two-machine flow shops, *International Journal of Production Economics,* No. 141, ISSN 09255273

► Yamamura, M. et al. (1992). Character-Preserving Genetic Algorithms for Traveling Salesman Problem, *Journal of Japanese Society for Artificial Intelligence*, Vol.7, No.6, pp1049-1059, ISSN 09128085

► Yu, S. et al. (1997). *System identification using dynamic ga based on numeric encoding,* Journal of Central South University of Technology, Volume 4, Issue 2, pp 128-131, Central South University of Technology, ISSN 1005-9784.