Towards Implementing Reactive Scheduling for Job Shop Problem

Zalmiyah Zakaria
Safaai Deris
Software Engineering Department
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia
<zalmiyah, safaai>@fsksm.utm.my

Abstract

Most of the research literature concerning scheduling concentrates on the static problems, i.e problems where all input data is known and does not change over time. However, the real world scheduling problems are very seldom static. Events like machine breakdown or bottleneck in some situation impossible to predict. Dynamic scheduling is a research field, which take into consideration uncertainty and dynamic changes in the real world scheduling problem. This paper gives an overview of the real problem occured in the filed of dynamic scheduling. Then we propose a hybrid genetic algorithm for solving the dynamic job shop problem.

Keywords

Dynamic scheduling, reactive scheduling, job shop scheduling, genetic algorithms

1.0 Introduction

Scheduling problem can be found in many different application areas, e.g. manufacturing, logistic, transportation, communication, sports, education, administration, etc. Main task of scheduling is the creation of schedules, which are temporal assignments of a set of activities to a set of resources subject to a set of constraints. Examples of scheduling constraints include deadlines (e.g., job *i* must be completed by time *t*), resource capacities (e.g., there are only two machine for drill), precedence constraints on the order of tasks (e.g., a leaf must be painted before it is assembled), and priorities on tasks (e.g., finish job *j* as soon as possible while meeting the other deadlines).

Many scheduling problems are difficult to solve [1]. It has been shown that many scheduling problems are NP-hard problem [2, 3, 4, 5, 6, 7, 8] - the time required to compute an optimal schedule increases exponentially with the size of the problem, meaning that with present-day algorithms even moderately sized problems cannot be solved to guaranteed optimality.

The rest of this paper is organized as follows. In section 2, the current issues that motivate the research on this area are discussed. In Section 3, a detailed description of the Job Shop Scheduling Problem (JSSP) is given. Section 4 summarizes the research done concerning JSSP. Section 5 and 6 discussed the previous genetic algorithms research aimed at solving the dynamic JSSP. Section 7 describes the current issues and challenges in this research area. In section 8, summarized the future plans.

2.0 Motivation

Basically there are two kinds of scheduling problems [9]. The first problem is static problem which related to the combinatorial nature of the problems, where it is difficult to find an optimal solution because it is impossible to consider all nodes in a large search space. This problem is also called generative in [10] and predictive in [11, 12]. While the second problem is dynamic problem which related to the dynamic nature of the problems, where variables and constraints which always change due to the development of an organization or emergence of certain type of events. This problem is also called revisions in [11] and reactive in [10, 12]. This problem is viewed as the reactive part of the system which monitors the execution of the schedule and copes with unexpected events (i.e., machine breakdowns, tool failures, order cancelation, due date changes, etc) [11].

The major criticism brought against the predictive mechanisms in practice is that the actual events on the shop floor can be considerably different compared to the one specified in the schedule due to the random interruptions (i.e., machine breakdowns, bottleneck, due date changes, order cancelations, etc.) [13, 14]. Thus an appropriate corrective action (or response) should be taken to improve the performance of the infeasible schedule [7, 10, 11, 12, 15, 16, 17]. Although reactive scheduling is of great importance in any scheduling system, most scheduling research has mainly focused on the construction of a good generative schedule from scratch without providing enough attention on the reactive control phase.

In industrial practice, the majority of scheduling systems address the reactive scheduling problem by making it the responsibility of the human scheduler to evaluate the implications of the unexpected events, and to adjust the generative schedule accordingly [10, 12]. However, the combinatorial complexity of the scheduling problem tends to overburden the human scheduler and may result in poor schedule performance.

Because of the dynamic environment Graves [18] stated that there is no scheduling problem but rather a rescheduling problem.

Responding to the dynamic factors immediately as they occur is also called real-time scheduling [13]. The initial schedule will be rescheduled to cope with the new conditions. This can also be called a time critical decision making process since the shop waits to receive the new schedule.

3.0 Definition of the JSSP

A $N \times M$ job shop scheduling problem, hereafter referred to as the JSSP, consists of N jobs and M machines [8]. A job j consists of a sequence of operations $O_j = (o_{j1}, o_{j2},\ldots,o_{jkj})$. Each operation o_{jl} is to be processed on a specific machine and has a specific processing time τ_{jl}. Each job has at most one operation on each machine (*capacity constraint*). The processing order of the operations in job j must be the order specified in the sequence O_j. These sequences are often called the technological constraints and also referred to as the *precedence constraint*. During processing each machine can process at most one operation at a time, and no preemption can take place; once processing of an operation has been started it must run until it has completed. In the following C_j will denote the end of processing time of the last operation of job j in a given schedule.

Some problems include a *due date* d_j for each job, a time by which the processing of the job is supposed to be finished, a *release time* r_j for each job, prior to which no processing of the job can be done, or a initial *setup time* s_m for each machine, prior to which no processing can be done on the machine.

A number of different objective functions exist for job shop problems. The most extensively researched is the *makespan* $C_{max} = max_{j\in\{1..N\}}(C_j)$, the time span needed to complete all operations of all jobs. However the makespan objective is not well-suited for scheduling on a rolling time horizon-basis (jobs arriving continuously over time), and since it does not include due dates. More realistic objectives include *total flowtime* $F = \Sigma^{N}_{j=1} C_j - r_j$, *summed lateness* $L_\Sigma = \Sigma^{N}_{j=1} C_j - d_j$, *summed tardiness* $T_\Sigma = \Sigma^{N}_{j=1}\ max\ (C_j - d_j, 0)$, *maximum lateness* $L_{max} = max_{j\in\{1..N\}}(C_j - d_j)$ and *maximum tardiness* $T_{max} = max\ (L_{max}, 0)$. All of these performance measures reflect schedule implementation cost and are to be minimised, i.e., a low performance measure equals a good schedule.

Table 1 : A 3 × 3 problem

job	Operations routing (processing time)		
1	1 (3)	2 (3)	3 (3)
2	1 (2)	3 (3)	2 (4)
3	2 (3)	1 (2)	3 (1)

An example of a 3 × 3 JSSP is given in Table 1. The data includes the routing of each job through each machine and the processing time for each operation (in parentheses). Figure 1 shows a solution for the problem represented by "Gantt-Chart".

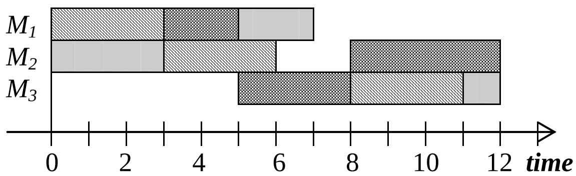

Figure 1: A schedule for a 3 x 3 JSSP instance

Based on the release times of jobs, JSSP can be classified as static or dynamic scheduling. In *static* JSSP, all jobs are ready to start at time zero. In *dynamic* JSSP, job release times are not fixed at a single point, that is, jobs arrive at various times. Dynamic JSSP can be further classified as deterministic or stochastic based on the manner of specification of the job release times. *Deterministic* JSSP assume that the job release times are known in advance. In *stochastic* JSSP, job release times are random variables and some or all parameters are uncertain [3, 5].

4.0 Related works

As discussed earlier, the majority of the published literature in the scheduling area deals with the task of schedule generation or predictive nature of the scheduling problems. The normally employed approaches for the solution of these problems are heuristic strategies [4]. Some of the most common techniques used are branch and bound [19], dispatching rules [20, 21], tabu search [22, 23, 24, 25, 26], simulated annealing [27, 28, 29] and genetic algorithms [2, 3, 5, 7, 8, 17, 30, 31, 32, 33]. In [34] and [35] we can found an extensive study about the main techniques that were applied since the year 1960s. The application of GA to scheduling problems has interested many researchers due to the fact that they seem to offer the ability to cope with the huge search spaces involved in optimizing schedules.

However, reactive scheduling is also important for the successful implementation of scheduling systems. A review on research papers that are related to reactive scheduling was given in [11]. This paper gives a short classification and a brief description about the existing studies concerning reactive scheduling.

Another popular approach to deal with reactive scheduling is knowledge-based system or expert system [14, 32, 36, 37, 38, 39, 40, 41, 42].

As stated earlier the common practice related to reactive scheduling in industrial practice is to assign human schedulers to repair the schedules using their knowledge and experience in the particular domain. This scenario shows that knowledge and experience are the most important elements to make the scheduling system become reactive because knowledge can provide information on where jobs are, where they need to go and what machine are up or down, etc.

A discussion on the knowledge-based reactive scheduling systems can be found in [34] and [43]. Cowling and Johansson [14] proposed a framework to use real time information to improve scheduling decisions, which allows the trade off between the quality of the revised schedule against the production disturbance which results from changing the planned schedule.

Shah et al. [44] developed knowledge based dynamic scheduling for production of parts in a steel plant. A rule base is used to handle the shared transporter, moving components and treated in sequence stations.

5.0 Dynamic JSSP

Dynamic problems have been considered on a *rolling time horizon basis*, in which the problem is solved by making a schedule for the part of the problem that is known. Processing of the jobs according to this schedule is then started, and as soon as information about new jobs arrive a new schedule incorporating the new jobs and the work not yet processed in the previous schedule is created.

Most research on scheduling has been focused mainly on optimizing one particular performance measure, like the use of resources, makespan or tardiness, normally reflecting some kind of cost. It is assumed that all problem data are known before scheduling has to take place and no change ever happens. However real world applications operate in dynamic environments frequently subject to several kinds of random occurrences and perturbations, such as new job arrivals, machine breakdowns, employees sickness, jobs cancellation and due date and time processing changes, causing that the original schedule becomes unfeasible.

Due to their dynamic nature, real scheduling problems have an additional complexity in relation to static ones. In many situations these problems, even for apparently simple situations, are hard to solve, i.e. the time required to compute an optimal solution increases exponentially with the size of the problem [6].

For such class of problems, the goal is no longer to find a single optimum, but rather to continuously adapt the solution to the changing environment. When a change in the environment happens rescheduling is needed, and the existence of a good near-optimal schedule, which is easy to modify will be in some situations preferable to an optimal, which cannot be modified.

The algorithms for dynamic scheduling should be able to manage any disruption of a schedule caused by changes in scheduling environment. Such changes can be classified in three major groups [16] :

- **Activity Changes**
  Request for new or extended activities can result in resource contention and inconsistency of a schedule. In long term scheduling introducing new activities can aim at improving the schedule efficiency and degree of resource utilization (e.g. leasing out some resource leads). In the short term scheduling activities are introduced as they arise (e.g. emergency service). Changes in activity duration and increased level of resource usage can occur.

- **Resource Changes**
  Primary reduction of resources (e.g. machine failure) can disrupt a schedule. Resource changes may be also requested to reduce the

cost of a schedule (e.g. machine utilization problems). Shorter term resource changes are usually connected with resource failure.

- **Temporal Changes**
  The most frequent form of temporal change is a contraction of schedule horizon. Long term temporal changes (e.g. changing a schedule in public transport for regularity) and short time changes (e.g. downstream effect of delayed aircraft or train) may also cause schedule inconsistency.

6.0 Genetic Algorithms (GA)

GA appeared around the end of the 1960s. Since Davis proposed the first GA-based technique to address scheduling problems in 1985 [44], GA have been widely used in the context of job shop scheduling problems (JSSP) [3, 4, 5, 17]. However, most of the works deal with optimisation of the scheduling problem in static environments, in which all jobs are ready to start at time zero, with the makespan objective. In dynamic JSSP, which are more realistic, jobs can arrive at some known (deterministic JSSP) or unknown (stochastic JSSP) future times. Further, the importance of each job can be different and the objective is more complex [3].

7.0 Issues and Challenges

Although scheduling is a well researched area, and numerous articles and books have been published, classical scheduling theory has been little used in real production environments [45]. It is believed that scheduling research has much to offer industry and commerce, but that more work is needed to address the ‘gap’ between scheduling theory and practice [14, 46]. One frequent assumption of scheduling theory, which rarely holds in practice, is that the scheduling environment is static. In recent years many authors [7, 10, 11, 12, 13, 14, 15, 16, 17, 46] have recognized that this is unlikely scenario in many manufacturing environment. In reality, schedules must be revised frequently in response to both instantaneous events, which occur without warning, and anticipated events where information is given in advance by, for example, process control computers or customers.

As a consequence, even though GA have previously been demonstrated to have an acceptable performance on job shop problems, it is still have not been adopted in standard manufacturing practice. For this reason, in recent years, academic research has attempted to consider real-life scheduling problems. Standard benchmark problems do not attract the attention of people in industry since practical scheduling problems are far more complex than the famous benchmark problems [4] that are still used in most research.

For the comprehensive comparison and summary of results that have been published for the Lawrence’s [47] and Fisher and Thompson’s [6] benchmark problems see [4].

However, a considerable number of recently published papers address real-life scheduling cases. Vieira et al. [48] described the development of a global scheduling system for a semiconductor test area. Gilkinson et al. [49] tackled the scheduling problem of a company that produces laminated paper and foil products. Hamada et al. [50] approached a complex scheduling problem in a steel-making company using a hybrid system based on evolutionary algorithms and expert systems. Shaw and Fleming [51] and Kumar and Srinivasan [52] proposed evolutionary computation methods for the solution of scheduling problems in companies that produce ready-chill meals and defense products, respectively. Sakawa et al. [53] considered the scheduling problem of a machining center using an evolutionary algorithm. Shah et al. [44] developed knowledge based dynamic scheduling for Steel Plant. Finally, Suh et al. 1998 [10] implemented ordering strategies for constraint satisfaction in steel industry. A scheduling expert system was developed to implement these strategies for the reactive adjustment of hot-rolling schedules in a hot strip mill.

8.0 Suggestion for further work

We propose to use GA with a knowledge-based scheduling to solve dynamic problem in the job shop scheduling problem.

GA was chosen since it is well suited to optimization problem and were proved successfully solve a number of problem that were difficult to solve with other methods [32]. We proposed to use knowledge because knowledge allows the use of global information to improve schedule decisions in the dynamic manufacturing environment.

In order to make this JSSP realistic to the real world problem, we will use the real data from spring manufacturing as a case study.

9.0 Conclusion

This paper described the actual problem happened in the job shop scheduling problem. It also discussed the previous work related to this area. Knowledge-GA is proposed to be developed in order to solve the dynamic problem in the real manufacturing environment.

10.0 Reference

[1] Parker R. G. (1995) Deterministic Scheduling Theory, London, Chapman & Hall.

[2] Yamada T and Nakano R. A Genetic Algorithm Applicable to Large-Scale Job Shop Problems. Parallel Problem Solving from Nature 2. (PPSN 2) 1992.

[3] Lin S., Goodman E.D. and Punch W. F. A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problems. International Conference of Genetic Algorithms (ICGA) 1997.

[4] Dimopoulos C. and Zalzala A.M.S. Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons. IEEE Transactions on Evolutionary Computation, Vol. 4, No. 2, July 2000 pp. 93 – 113.

[5] Madureira A. M., Ramos C. and Silva S. D. C. A Genetic Approach for Dynamic Job-Shop Scheduling Problems. 4th MetaHeuristics International Conference (MIC'2001), Portugal, 2001.

[6] Blazewicz J, Ecker K.H., Pesch E, Schmidt G. and Weglarz J. Scheduling Computer and Manufacturing Processes, Springer, Berlin, 2001.

[7] Jensen M. T. Generating Robust and Flexible Job Shop Schedules using Genetic Algorithms. IEEE Transactions on Evolutionary Computation, Vol. 7, no 3, June 2003, pp. 275-288.

[8] Braune R., Wagner S. and Affenzeller M. Applying Genetic Algorithms to the Optimization of Production Planning. Real-World Manufacturing Environment, Cybernetics and Systems, 2004.

[9] Deris S., Omatu S., Ohta H. and Saad P. Incorporating constraint propagation in genetic algorithm for university timetable planning. Journal of the Engineering Application of Artificial Intelligence, Vol. 12, 1999, pp. 241-253.

[10] Suh M. S., Lee A., Lee Y. J. and Ko Y. K. Evaluation of ordering strategies for constraint satisfaction reactive scheduling. Decision Support Systems, Vol. 22, Issue 2, Feb. 1998, pp. 187-197.

[11] Sabuncuoglu I. and Bayiz M. Analysis of reactive scheduling problems in a job shop environment. European Journal of Operational Research, Vol. 126, Issue 3, Nov. 2000, pp. 567-586.

[12] Sauer J. Planning and Scheduling - An Overview. in: Hotz, L., Krebs, T. (Eds.): Planen und Konfigurieren (PuK-2003), Proceedings des Workshops zur KI 2003, Hamburg, 2003, pp. 158-161.

[13] Akturk, M. S. and Gorgulu, E., Match-up scheduling under a machine breakdown, European Journal of Operational Research, Vol. 112, Issue 1, Jan. 1999, pp. 81-97.

[14] Cowling P. and Johansson M., Using real time information for effective dynamic scheduling, European Journal of Operational Research, Vol. 139, Issue 2, June 2002, pp. 230-244.

[15] Dorn, J. Case-based reactive scheduling. in Roger Kerr and Elisabeth Szelke (eds) Artificial Intelligence in Reactive Scheduling. London: Chapman & Hall. 1995, pp. 32-50.

[16] Kocjan W. Dynamic scheduling: State of the art report. Technical Report T2002:28, SICS, 2002.

[17] Vazquez M. and Whitley L. D. A comparison of genetic algorithms for the dynamic job shop scheduling problem. Proceedings of GECCO-2000, 2000, pp. 169-178.

[18] Graves S.C. A review of production scheduling, Operations Research, Vol. 29, Issue 4, 1981, pp. 646-675.

[19] Carlier J. and Pinson E. An Algorithm for Solving the Job-Shop Problem. Management Science, Vol. 35, 1989, pp. 164-176.

[20] Baker K. R. (1974) Introduction to Sequencing and Scheduling, New York: John Wiley.

[21] Lodree E. J., Jang W., Klein C. M. A new rule for minimizing the number of tardy jobs in dynamic flow shops. European Journal of Operational Research 159, 2004, 258-263.

[22] Hurink J. and Knust S., A tabu search algorithm for scheduling a single robot in a job-shop environment, Discrete Applied Mathematics, Vol. 119, Issues 1-2, June 2002, pp. 181-203.

[23] Watson J. P., Beck J. C., Howe A. E. and Whitley L. D. Problem difficulty for tabu search in job-shop scheduling, Artificial Intelligence, Vol. 143, Issue 2, Feb. 2003, pp. 189-217.

[24] Carotenuto P., Giordani S., Ricciardelli S. and Rismondo S., A tabu search approach for scheduling hazmat shipments. Computers & Operations Research, In Press, Corrected Proof, Available online 22 July 2005.

[25] Liu S.Q., Ong H.L. and Ng K.M., A fast tabu search algorithm for the group shop scheduling problem. Advances in Engineering Software, Vol. 36, Issue 8, Aug. 2005, pp. 533-539.

[26] Xu J., Sohoni M., McCleery M., Bailey T. G., A dynamic neighborhood based tabu search algorithm for real-world flight instructor scheduling problems. European Journal of Operational Research, Vol. 169, 2006, pp. 978–993.

[27] Mamalis A. G. and Malagardis I. Determination of due dates in job shop scheduling by simulated annealing. Computer Integrated Manufacturing Systems, Vol. 9, Issue 2, May 1996, pp. 65-72.

[28] Satake T., Morikawa K., Takahashi K. and Nakamura N. Simulated annealing approach for minimizing the makespan of the general job-shop. International Journal of Production Economics, Vol. 60-61, April 1999, pp. 515-522.

[29] Aydin M. E. and Fogarty T. C. Simulated annealing with evolutionary processes in job shop scheduling.

Evolutionary Methods for Design, Optimisation and Control, (Proc. of EUROGEN 2001, Athens, 19-21 September) CIMNE, Barcelona, 2002

[30] Fang H.L., Ross P., Corne D., A promising genetic algorithm approach to job-shop scheduling, rescheduling, and open-shop scheduling problems. Proceedings of the Fifth International Conference on Genetic Algorithms, 1993, pp. 375–382.

[31] Bierwirth C. and Mattfeld D.C. Production scheduling and rescheduling with genetic algorithms, Evolutionary Computation, Vol. 7, Issue 1, 1999, pp. 1–17.

[32] Varela R., Vela C.R., Puente J. and Gomez A. A knowledge-based evolutionary strategy for scheduling problems with bottlenecks. European Journal of Operational Research, Vol. 145, 2003, pp. 57–71.

[33] Goncalves J. F, de Magalhaes Mendes J. J., Resende M. G. C. A hybrid genetic algorithm for the job shop scheduling problem. European Journal of Operational Research, Vol. 167, 2005, pp. 77–95.

[34] Blazewicz J., Domschke W. and Pesch E. The job shop scheduling problem: Conventional and new solution techniques. European Journal of Operational Research, Vol. 93, 1996, pp. 1–33.

[35] Jain A.S., Meeran S., Deterministic job-shop scheduling: Past, present and future, European Journal of Operational Research, Vol. 113, 1999, pp. 390–434.

[36] Shah V. C., Madey G. R. and Mehrez A. A methodology for knowledge-based scheduling decision support, Omega, Vol. 20, Issues 5-6, Sept.-Nov. 1992, pp. 679-703.

[37] Collinot A. and Le Pape C. Adapting the behavior of a job-shop scheduling system. Decision Support Systems, Vol. 7, Issue 4, Nov. 1991, pp. 341-353.

[38] Ress D. A. and Currie K. R. Development of an expert system for scheduling work content in a job shop environment. Computers & Industrial Engineering, Vol. 25, Issue 1-4, Sept. 1993, pp. 131-134.

[39] Charalambous O. and Hindi K. S. A knowledge based job-shop scheduling system with controlled backtracking. Computers & Industrial Engineering, Vol. 24, Issue 3, July 1993, pp. 391-400.

[40] Miyashita K. and Sycara K. CABINS: a framework of knowledge acquisition and iterative revision for schedule improvement and reactive repair. Artificial Intelligence, Vol. 76, Issues 1-2, July 1995, pp. 377-426.

[41] Zhang Y. and Chen H. A knowledge-based dynamic job-scheduling in low-volume / high-variety manufacturing. Artificial Intelligence in Engineering, Vol. 13, Issue 3, July 1999, pp. 241-249.

[42] Henning G. P. and Cerda J. Knowledge-based predictive and reactive scheduling in industrial environments. Computers & Chemical Engineering, Vol. 24, Issues 9-10, Oct. 2000, pp. 2315-2338.

[43] Szelke E. and Kerr R.M. Knowledge-based reactive scheduling, Production Planning and Control, Vol. 5, Issue 2, 1994, pp. 124-145.

[44] Shah M.J., Damian R., Silverman J. Knowledge Based Dynamic Scheduling in a Steel Plant. Proceedings of the 6th International Conference on Artificial Intelligence for Industrial Applications, St. Barbara, 1990, pp. 108-113.

[45] Stoop P. and Wiers V. The complexity of scheduling in practice. International Journal of Operations & Production Management, Vol. 16 No. 10, 1996, pp. 37-53.

[46] Aytug H., Lawley M. A., McKay K., Mohan S. and Uzsoy R. Executing production schedules in the face of uncertainties: A review and some future directions. European Journal of Operational Research, Vol. 161, Issue 1, Feb. 2005, pp. 86-110.

[47] Lawrence S., Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques. Pittsburgh, PA: GSIA, Carnegie Mellon Univ., 1984.

[48] Vieira G.E., Herrmann J.W. and Lin E., Rescheduling manufacturing systems: a framework of strategies, policies, and methods, Journal of Scheduling, Vol. 6, No. 1, pp. 35-58, 2003.

[49] Gilkinson J. C., Rabelo L. C., and Bush B. O., A real-world scheduling problem using genetic algorithms. Computers & Industrial Engineering, Vol. 29, Issue 1–4, 1995, pp. 177–181.

[50] Hamada K., Baba T., Sato K., and Yufu M. Hybridizing a genetic algorithm with rule-based reasoning for production planning. IEEE Expert, Vol. 10, No. 5, 1995, pp. 60–67.

[51] Shaw K. J. and Flemming P. J. Including real-life preferences in genetic algorithms to improve optimization of production schedules. Proc. Conf. Genetic Algorithms Eng. Syst.: Innovations and Appl.. Stevenage, U.K.: IEE, 1997, pp. 239–244. IEE Conf. Publ. 446.

[52] Kumar N. S. H. and Srinivasan G. A genetic algorithm for job-shop scheduling - A case study. Computer Industry, Vol. 31, no. 2, 1996, pp. 155–160.

[53] Sakawa M., Kato K. and Mori T. Flexible scheduling in a machining center through genetic algorithms. Computers & Industrial Engineering, Vol. 30, No. 4, September 1996, pp. 931-940.

[54] Ovacik I. M, Uzsoy R. Exploiting shop floor status information to schedule complex job shops. Journal of Manufacturing Systems. Dearborn, Vol. 13, Issue 2, 1994, pp. 73-84.