Production Systems and Information Engineering
Volume 5 (2009), pp. 81-94.

SOLVING MULTI-OBJECTIVE PRODUCTION SCHEDULING PROBLEMS USING A NEW APPROACH

GYULA KULCSÁR
University of Miskolc, Hungary
Department of Information Engineering
kulcsar@ait.iit.uni-miskolc.hu

MÓNIKA KULCSÁRNÉ FORRAI
University of Miskolc, Hungary
Department of Automation
kulcsfm@mazsola.iit.uni-miskolc.hu

[Received March 2009 and accepted April 2009]

Abstract. Most companies use a proactive approach to schedule production orders and jobs at shop floor level. To make a near-optimal schedule for a shop with different types of machines and many operations is a very important but complicated task because of the very large number of alternative solution in the searching space. Advanced scheduling models, good heuristics and fast improving algorithms are required to satisfy constraints and to optimize production performances. The aim of the paper is to outline a new modelling and solving approach connected to discrete production scheduling and rescheduling.

Keywords: scheduling, multi-objective optimization, simulation, production

1. Introduction

Today, production engineering and management utilize more and more computer-integrated application systems to support decision making. Software systems have been applied to manage discrete production processes. These can be classified into four hierarchical groups according to the supported fields: (1) Enterprise Resources Planning (ERP), (2) Computer Aided Production Engineering (CAPE), (3) Manufacturing Execution Systems (MES) and (4) Manufacturing Automation (MA). This paper focuses on the detailed scheduling function of Manufacturing Execution Systems. At MES level, the main purpose of the fine scheduler is to initiate a detailed schedule to meet the master plan defined at ERP level. The scheduler gets the current data of dependent orders, products, resource environment and others technological constraints (tools, operations, buffers, materials handling and so on). The shop floor management defines the production goals and their priorities. Obviously, the management may declare different goals at different times. The scheduler has to provide a feasible sequence of jobs which meets the management's goal. The result of the scheduling is a detailed production programme which declares the releasing sequence of jobs, assigns all the necessary resources to them and proposes the starting time of operations. It must not break any of the hard constraints but has to meet the predefined goals. The computation time of the scheduling process is also an important issue especially with a large number of internal orders, jobs, operations, resources, technological variants and constraints.

In discrete manufacturing, series of goods are produced. A series (batch or lot) can include highly different number of pieces, from a single product (e.g. special part, complex or unique equipment) to thousands or millions of the same product (simple parts or goods). In discrete manufacturing operations are executed on discrete, separated machines and workplaces. Depending on the arrangement of machines, buffers and transportation devices, manufacturing systems may have a line type or group type layout. In essence the execution of the operations for mass production or customized mass production requires the exact predefinition of the routing of the operations. This kind of model is called flow shop (FS) model.

In a FS model there are machines in a theoretical line structure, placed one after the other in the order predefined by the technology. The model is highly influenced by the presence of buffers between machines. If the capacity of the buffers is zero, then the system is a strictly synchronized transfer system, othervise the system is called asynchronous transfer line system. A permutation flow shop model is a special variant of the FS models, it means that the job-queues in front of each machine operate according to the FIFO (First In First Out) principle.

In the literature, different advanced variants of the classical FS models can be found. One of the main groups of these models is the flexible flow shop (FFS) scheme [1, 4, 6, 7, 8, 9, 14, 16]. The FFS environment consists of stages which represent the fundamental operation-type units of the manufacturing / assembly system. At each stage one or more identical machines work in parallel. Each job has to be processed at each stage on any of the parallel concurrent machines. In respect of production performance, both the allocation of machines and the order of jobs are of great importance. Lots of flexible flow shop models are known in the literature, but most of them use only one performance measure (objective function). Usually the latest finishing time (makespan) of the released jobs appears as goal function of optimization for make to stock (MTS) manufacturing. Frequently one of objective functions related to due date plays the main role in scheduling models for supporting make to order (MTO) manufacturing. Only a few of the models deals with multi-objective cases which are more important in flexible and agile manufacturing. According to the customized mass production (CMP) paradigm the firms plan their production partially for external direct orders. Additionally, to reach better delivery capabilities they make forecasts for manufacturing to make components, master units or semi-finished products for stock. The flexible scheduling models – known in the literature – do not meet all the requirements of CMP to the expected extent. The existing models often disregard the machine processing abilities, alternative technological routes, limited availability time of the machines, limited buffer capacities, shared machine tools, so an improvement and extension of flexible flow shop models are required.

2. A New Advanced Fine Scheduling Approach

2.1. An Extended Flexible Flow Shop Model (EFFS)

The discrete manufacturing process examined produces various consumer goods. By means of forecasting tools which consider external orders, market trends, seasonal characteristics a set of internal orders has been created by the production planners. Each order defines the required number of identical products of a certain product type, which should be manufactured by the predefined time. The logistic unit is the palette at the shop floor level, which can take one or more products. Internal orders consist of one or more (i.e. whole number of) palettes. Depending on the product type, palettes carry a predefined number of identical products. **O**rders can be considered the set of palettes to manufacture, where the number of palettes depends on the ordered product quantity and the capacity of the palettes. The model being shown in this section applies manufacturing / assembly machine objects (individual machines and/or machine lines). Machine lines perform several technological steps (TS). Each TS means a sequence of elementary operations and cannot be interrupted. Consequently a TS is the smallest allocation unit during the scheduling. A job means one or more palette of an internal order with technological steps to be executed in a predefined sequence. The nature of flexible manufacturing is that same goods can be manufactured using alternative materials, components, machines or technological routes. The capacity of the buffers placed among the machines can be zero, limited or not limited. The limit size of a buffer may depend on the product types.

In EFFS model, every machine can be characterized by product sequence dependent setup times, availability time frames, various production rates depending on product types, and capability for performing a single step or a sequence of steps for certain products. Machines can be arranged into machine groups according to processing ability. A machine group is a set of machines that can execute the same execution step (sequence of technological steps). This point of view, a given final product or a semi-finished product can be produced differently using different sequences of machine groups which the required components are taken through. The flow shop nature of the model means that each execution route may consists of one or more execution steps, but the common part of the execution steps has to be an empty set (overlapped technological step is not allowed). Moreover the sequence of technological steps is determined by a strict direction and an execution step has to be included in all the technological steps which are between the first and last steps of the machine.

In order to formulate the new class of scheduling problems described above, the well-known formal specification $\alpha|\beta|\gamma$ is used, where α denotes machine environment descriptors, β denotes processing characteristics and constraints, and γ denotes the list of objective functions. An extended Flexible Flow Shop (EFFS) scheduling model can be described as follows:

$$\mathrm{Fx}, \mathrm{M_g}, \mathrm{Q_{i,m}}, \mathrm{Set_{i,j,m}}, \mathrm{Cal_m}, \mathrm{B_{m,p}}, \mathrm{TR_{m,n}} \left| \mathrm{R_i}, \mathrm{D_i}, \mathrm{Exe_i}, \mathrm{A_i} \right| \mathrm{f_1}, \mathrm{f_2}, \ldots, \mathrm{f_K} \tag{2.1}$$

$$\alpha = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7\} \tag{2.2}$$

α_1: Type of operation (technology step) sequence at shop floor level. *F*: Flow Shop, *x*: the maximum number of operations.

α_2: Type of machine environment. M_g: group of multi-purpose machines which can execute one or more operations in a given sequence. Each machine group can include distinct parallel machines.

α_3: Type of alternative machines. $Q_{i,m}$: unrelated parallel machines with job dependent production rates.

α_4: Type of machine setups. $Set_{i,j,m}$: job sequence and machine dependent setup times.

α_5: Special resource constraint. Cal_m: machine availability time intervals.

α_6: Special buffer constraint. $B_{m,p}$: product type dependent capacity of the buffer placed in front of machines.

α_7: Transportation time. $TR_{m,n}$: job travelling time between given machines.

$$\beta = \{\beta_1, \beta_2, \beta_3, \beta_4\} \tag{2.3}$$

β_1: Constrained released time of jobs. R_i: the earliest start time of jobs.

β_2: Constrained due date of jobs. D_i: the latest completion time of jobs.

β_3: Type of manufacturing processes. Exe_i: required type and sequence of technology steps for jobs.

β_4: Constrained resource assignments. A_i: set of suitable machines for jobs.

$$\gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_K\} \tag{2.4}$$

The goodness and quality of the schedule can be evaluated using the numerical result of the objective function. Some examples are listed in Table 1. Real manufacturing environments may require various objective functions declared.

The extended flexible flow shop scheduling problem is difficult to solve because of its combinatorial nature. The model inherits the difficulties of the classical flow shop and the flexible flow shop models. Additionally, numerous odd features appear because of special extensions.

Table 1. Typical objective functions for detailed scheduling

C_{max}	Finishing time of last job (makespan) to be min.
L_{max}	Max lateness to be min.
T_{max}	Max tardiness to be min.
S_{max}	Max square distance of differences to due dates to be min.
$\sum(C_i-r_i)$	Sum of throughput times to be min.
$\sum L_i$	Sum of lateness times to be min.
$\sum T_i$	Sum of tardiness times to be min.
$\sum U_m$	Sum of machine utilization to be max.
$\sum v_i(C_i-r_i)$	Weighted sum of flow times to be min.
$\sum v_i L_i$	Weighted sum of lateness times to be min.
$\sum v_i T_i$	Weighted sum of tardiness times to be min.
N_{WIP}	Average number of work in progress to be min.
N_{SET}	Number of setups to be min.
$\sum T_{SET}$	Sum of setup times to be min.

2.2. Comparison of Solutions Based on Relative Quality

In the literature, different approaches can be found considering multi-objective scheduling problems, as they are surveyed i.e. in [3, 11]. In this section a new approach will be shown. According to our proposal the relative goodness of a solution is more important than its absolute goodness. The basis of the approach is the following: the relative goodness of the selected solution is measured by comparing it with another solution in the feasible solution space.

Let S be the search space under consideration. It is the set of all possible solutions of the problem. Suppose that a number of objective functions $f_1, \ldots, f_K$ is given such that:

$$f_k : S \to \Re^+ \cup \{0\}, \ \forall k \in \{1,2,\ldots,K\}. \tag{2.5}$$

The problem is to find an $s \in S$ that minimizes every $f_k(s)$. This is known as a multi-objective optimization problem. In the majority of cases, it is not possible to find a solution to a multi-objective optimization problem. Successfully minimizing one of the component objective functions will typically increase the value of another one. So we must find solutions that represent a compromise between the various criteria used to evaluate the quality of solutions.

Let s_x, $s_y \in S$ be two solutions. The function F is defined to express the quality of s_y compared to s_x as a real number:

$$F : S^2 \to \Re\,,\; F(s_x, s_y) = \sum_{k=1}^{K} (w_k \cdot D(f_k(s_x), f_k(s_y)))\,. \tag{2.6}$$

The definition of function D is the following:

$$D : \Re^2 \to \Re\,,\; a, b \in \Re\,,\; D(a,b) = \begin{cases} 0, \mathit{if}\ \max(a,b) = 0 \\ \dfrac{b-a}{\max(a,b)} \cdot 100,\ \mathit{otherwise} \end{cases}. \tag{2.7}$$

The *max(a,b)* used in (2.7) denotes an operator:

$$\max : \Re^2 \to \Re\,,\; \max(a,b) = \begin{cases} a,\ \mathit{if}\ a > b \\ b,\ \mathit{otherwise} \end{cases}. \tag{2.8}$$

Moreover, coefficient w_k which is an integer value within range *[0, 1, …, W]* is used in order to express the importance of any component objective function f_k. It is allowed that the decision maker sets the actual priority of each objective function independently.

Using the function F, two solutions s_x, $s_y \in S$ are compared as follows:

s_x is a better solution than s_y ($s_x < s_y$ is true) if

$$F(s_x, s_y) > 0\,. \tag{2.9}$$

s_x and s_y are equal ($s_x = s_y$ is true) if

$$F(s_x, s_y) = 0\,. \tag{2.10}$$

s_x is a worse solution than s_y ($s_x > s_y$ is true) if

$$F(s_x, s_y) < 0\,. \tag{2.11}$$

These definitions of the relational operators are suitable for applying in meta-heuristics like taboo search, simulated annealing and genetic algorithms to solve multi-objective combinatorial optimization problems.

In this model, the *max* operator (2.8), which means the basis of the comparison, may be replaced by a general function *g* but the essence of the proposed approach does not change. The relative qualification of two solutions remains if the new function *g* is characterized by the following simple feature (2.12):

$$g : \Re^2 \rightarrow \Re\,,\ g(a,b) = g(b,a)\,. \tag{2.12}$$

The mathematical model described above was developed to establish a method for managing the objective functions and evaluating the relative quality of the feasible solutions by comparing them to each other. The model can be widely used for solving multi-objective combinatorial optimization problems which include objective functions characterised by dynamically varying importance, different dimension and value range.

2.3. Numerical Simulation and Evaluation of Schedules

Manufacturing processes can be characterized by general state variables. In respect of production management the most important three state variables for measuring performance are as follows: (1) stock level, (2) capacity utilization and (3) readiness for delivery [12]. For solving scheduling problems we use fast computer simulation to evaluate the schedules. It considers the availability of the individual machines for a given time window and the required setup times between the series. Using the machine-job assignments the processing time of tasks can be calculated. For each job and each internal order the starting and finishing time can be defined by using the time data of all the tasks. By means of the simulation the objective functions can be evaluated as well.

The proposed algorithm means numerical simulation of the production to calculate the time data of the execution of tasks. Inputs of the simulation consist of jobs, machines, their assignments, sequences of jobs on machines, buffer capacities, abilities of machines, availabilities of machines, transportation time of jobs between machines. Simulation of a given job on an intermediate machine requires, among other things, the completion time of the job on the previous machine and the previous job on the machine, moreover the shop floor environment has lots of junctions of the possible routes. So it has to define the sequence of tasks in which the calculation can be performed correctly. To satisfy this requirement we developed a fast process oriented algorithm that works in an event driven way.

The time values of a given job (task) on an assigned machine are mainly determined by (1) the constraint start time of the job (in point of view of components availabilities), (2) completion time of the job on the previous machine,

(3) completion time of the previous job on the machine, (4) transportation time of the job taken from the previous machine, (5) setup time of the job on the machine, (6) the availability of the machine (availability time frames and product dependent production intensities), (6) actual state of the buffer at back of machine and (7) availability of tools needed (which can be shared).

The numerical tracking of the product-palettes supplies the time data of the manufacturing steps such as starting time, setup time, processing time and finishing time of tasks, jobs and internal orders. The simulator extends the predefined schedule (job-resource assignments and job sequences on machines) to a fine schedule by calculating and assigning time data. Consequently the simulation is able to transform the original searching space into a reduced space by solving the timing problem.

2.4. Integrated Fine Scheduling and Rescheduling Software

Meta-heuristics (i.e.: genetic algorithms, simulated annealing and taboo search) are becoming more and more successful methods for optimization problems that are too complex to be solved using deterministic techniques [3, 7, 10, 11, 15]. In general, the scheduling task consists of batching, assigning, sequencing and timing because of the complexity of the problem. We developed a new integrated approach to solve all these sub-problems as a whole without decomposition. In this approach, all the issues are answered simultaneously (Figure 1).

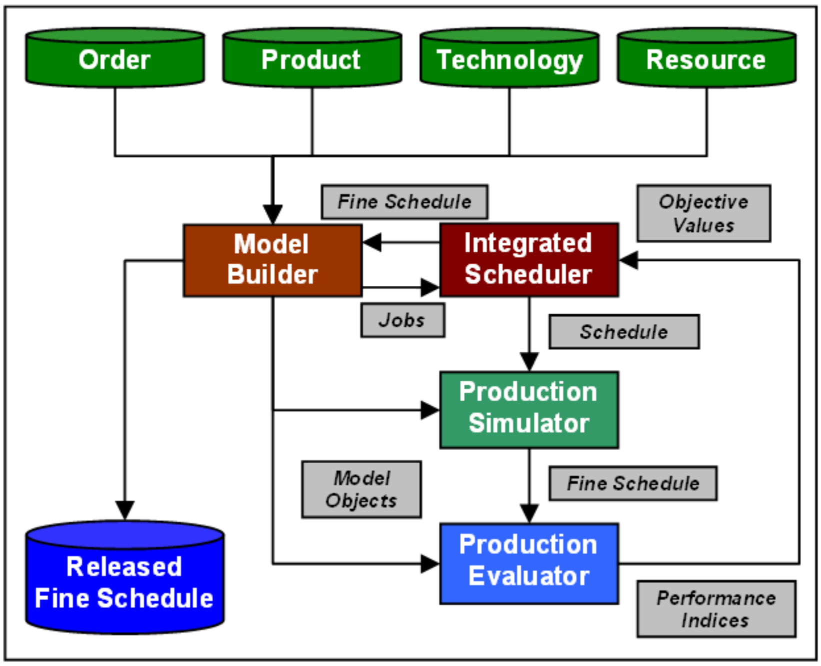

Figure 1. Integrated fine scheduling approach

The new integrated approach based method supports the decision making of joining and/or dividing production orders; the calculation of the manufacturing lot sizes dynamically; the selection of the alternative technological routes; the allocation of machine resources; the definition of manufacturing tasks and the scheduling of its execution processes. This method uses heuristic algorithms, searching techniques and problem space transformation based on discrete events type simulation.

To accelerate the computation an indexed data model has been elaborated. The data structure supports the association of two or more different types of arrays. The model builder creates the full indexed data model which includes the possible technology and resource alternatives.

In the approach applied the product-pallet plays the role of the basic scheduling unit. Each production order consists of pallets that mean individual jobs (one or more pieces with execution steps required). The production batch sizes are formed dynamically by scheduling the jobs on machines.

In order to create a detailed schedule for the production of each internal order, it is necessary for each job: (1) to be assigned to one of the possible routes, (2) to be assigned to one of the possible machines at each possible machine group according to selected route, (3) to fix its position in the queue of each selected machine, and (4) to fix its starting time on each selected machine.

The solving algorithms are integrated into a scheduler engine. Two classes of heuristic algorithms are used in two phases. In the first phase, constructive algorithms based on combined heuristic priority rules create good initial solutions. In the second phase, iterative searching algorithms improve the best solution according to the multiple objectives. The method focuses on creating near-optimal feasible schedules considering multiple objectives and it is based on a special taboo search variant. A certain number of neighbours of the current schedule are generated at random successively by using priority controlled neighbouring operators. These operators create new feasible schedules by modifying resource allocations and job sequences. It is not necessary to check the feasibility of the generated solutions because the neighbouring operators make valid modifications by choosing allowed alternatives from the indexed model structure. Moreover, an advanced structure of taboo-list is used. Taboo-list contains the schedules that have been visited in the recent past (less than a given number of moves ago). Schedules in the taboo-list are excluded from the neighbourhood of the actual solution.

The objective functions concerning schedules are evaluated by the production simulator which represents the discrete production environment (machines with their capabilities, buffers with their capacities and others technological constraints). The production evaluator measures the performance of the fine schedules by calculating management indices based on job, order and machine data. The mathematical model proposed in Section 2.2 for relative qualification is used for comparing the generated schedules according to multiple objectives. The best schedule becomes the initial solution of the next loop of the searching algorithm.

When the scheduling process has been finished or stopped by the user, the current best schedule is returned.

In managing real production systems, different types of uncertainty may occur e.g. machine failure or breakdown, missing material or components, under-estimation of processing time, job priority or due date changes and so on. Different rescheduling methods can be used according to the effects of the unexpected events: time shift rescheduling, partial rescheduling or complete rescheduling [2, 13]. Rescheduling is a process of updating an existing production schedule in response to disruptions or creating a new one if the current schedule has become infeasible.

Our approach is able to solve rescheduling tasks using multi-objective searching algorithms similarly to predictive scheduling (Figure 2). The aim of rescheduling is to find a schedule, which (1) considers the modified circumstances, (2) is near-optimal according to some predefined criterion and (3) is as close as possible to the original one.

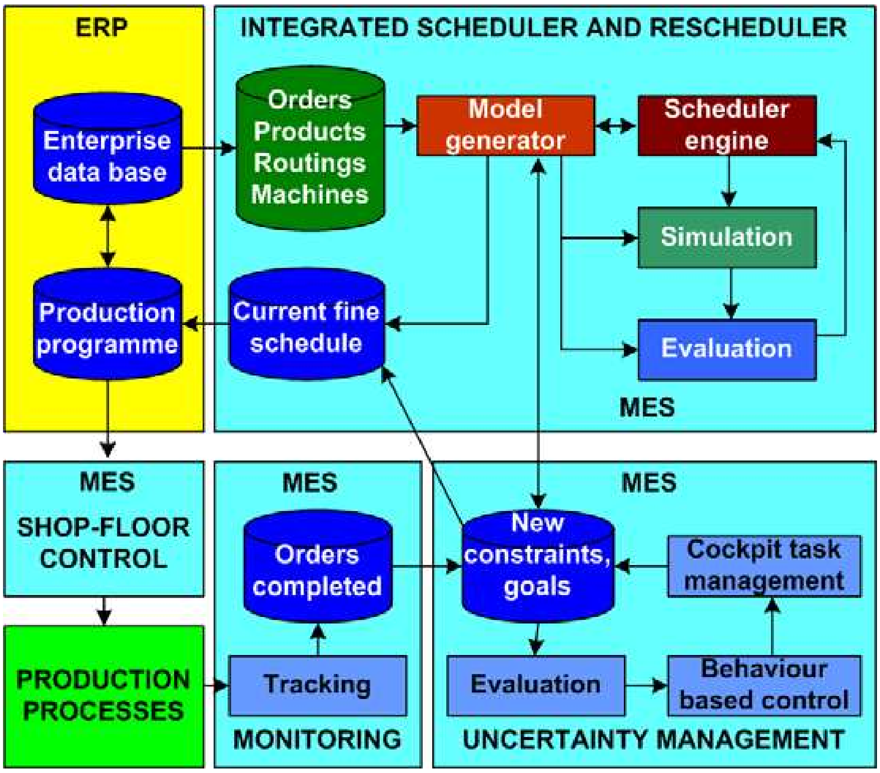

Figure 2. Integrated scheduling and rescheduling system

It is required of rescheduling methods to consider new demands that are added to the predictive scheduling problem. The last released schedule appears as a new input element of the rescheduling system and it is very important to preserve this initial schedule as much as possible to maintain the system stability. For this purpose we defined new qualitative indices (i.e. related to setup and due date) for supporting the comparison of schedule changes. We consider a great number of special rescheduling constraints. Some examples are as follows: All jobs which are already finished when the rescheduling process starts are not changeable but can affect the other jobs and orders. The manufacturing tasks of jobs running at the starting time of the rescheduling process must not be interrupted and their possible execution route and parallel machines (and other alternatives) can differ from the original possibilities. Finished or running jobs on resources are known therefore they have to be considered in the future. All production orders starting after the rescheduling process can be considered in their original status. To satisfy these constraints the software uses freezing techniques. The main classes of these techniques are as follows: (1) to freeze jobs, (2) to freeze internal orders and (3) to freeze machines. The advanced functionalities of the software help to satisfy the requirements of shop floor control in practise.

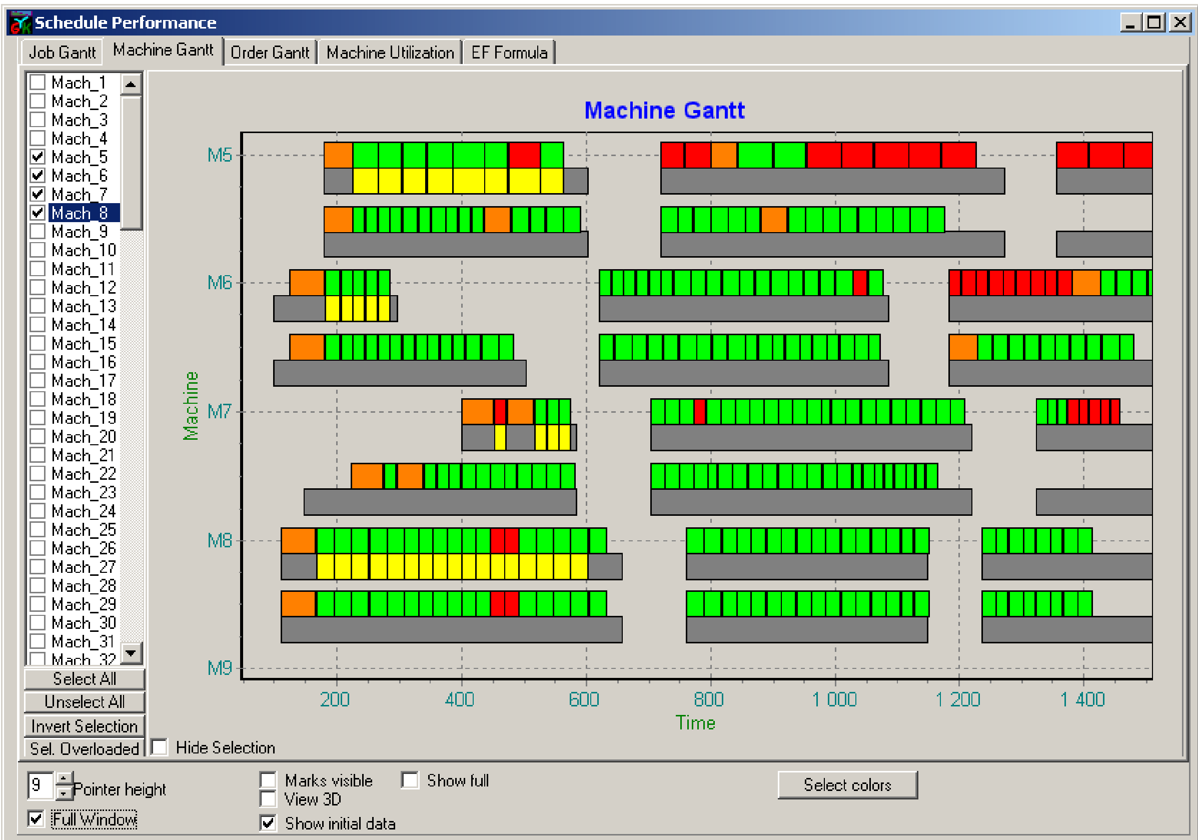

Figure 3. Visualization of the actual and planned status of the jobs

To increase the flexibility and effectiveness of the scheduling process, an advanced software module for supporting the user interactions has been developed. A graphical user interface provides useful charts, diagrams, tables and reports to show aggregate and detailed information of the production fine schedule (i.e. Figure 3). The scheduling process is also scrutinized and checked on the screen, the user can modify at runtime the control priority and parameters of searching algorithms (Figure 4). In addition, the user can suspend the automatic process and edit the actual schedule by using the available operation tools manually (Figure 5). Similarly to the neighbouring operators, the usage of the manual planning tools can produce only valid and feasible solutions.

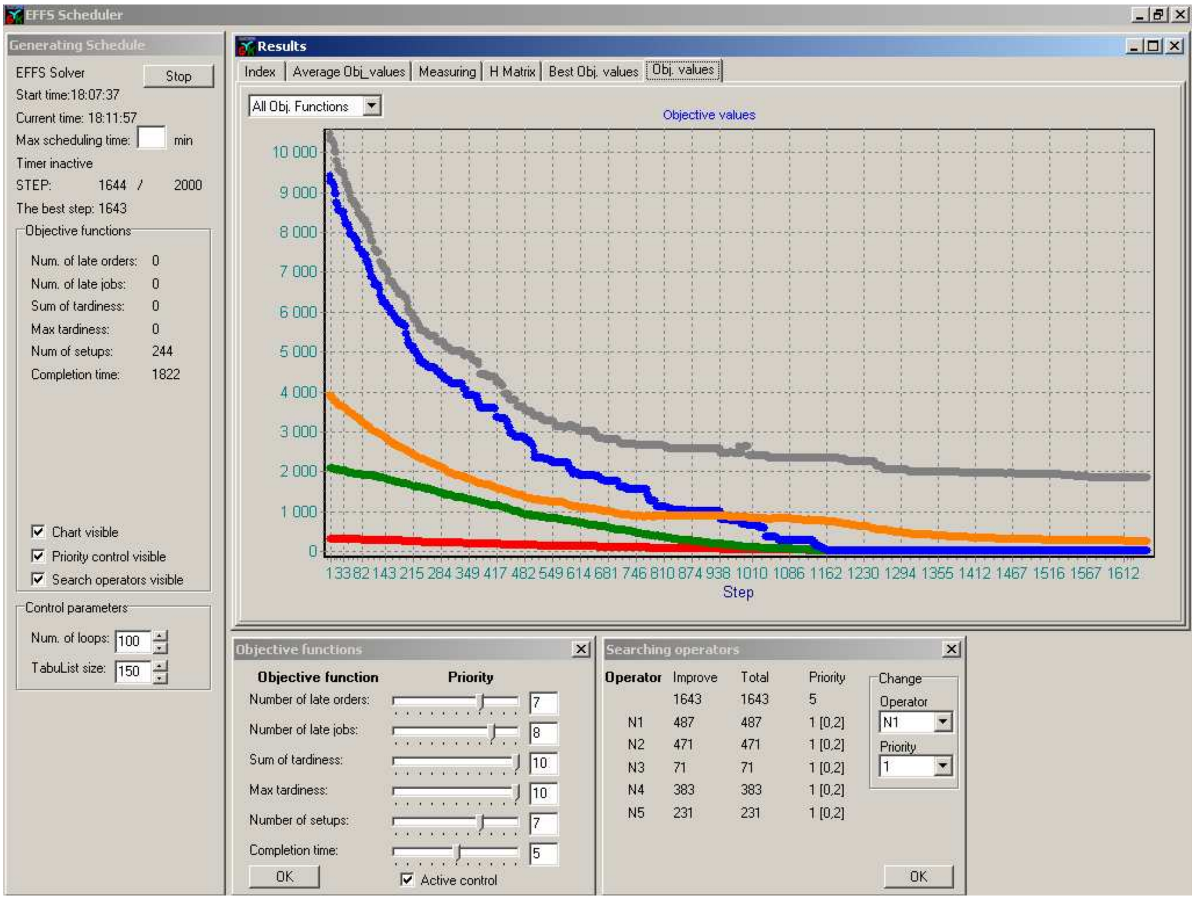

Figure 4. A typical user interface of the scheduling system

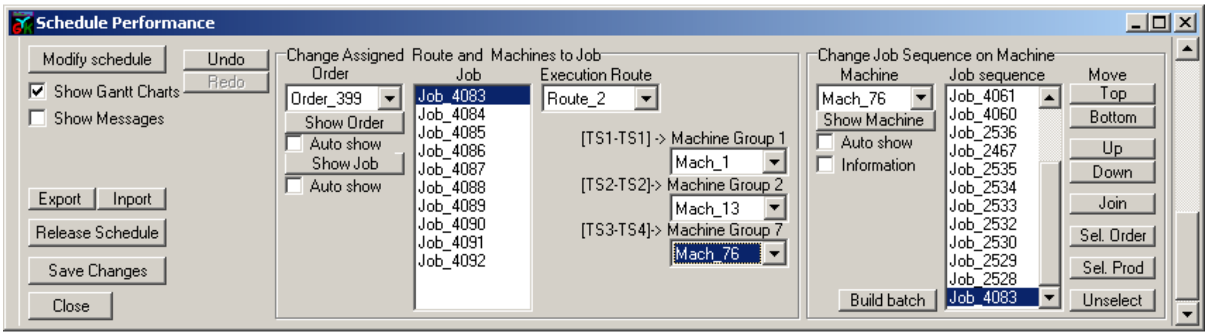

Figure 5. Operation tools for editing schedules manually

The running results produced on sample tasks show that the EFFS model developed and its solution methods are suitable for supporting various production planning and control tasks that fit in the defined category of tasks. Based on the results of the tests which are executed on small size problem instances, we can say that the method is able to find the optimal solution. The results are validated by comparison with the result of an optimal method based on enumeration technique. The method is able to solve large size problems effectively in a reasonable amount of time [5]. Demo version of the implemented software with built-in problem generator and some input data are available on the Web at the following address: http://ait.iit.uni-miskolc.hu/~kulcsar/EFFS_Sch_Demo.zip.

3. Conclusions

The paper describes the proposition and application of a new method for solving multi-objective scheduling and rescheduling problems. It is based on new interpretation and usage of relational operators for comparing quality of schedules in searching algorithms. After developing the software prototype, the concept is successfully tested on extended flexible flow shop scheduling and rescheduling problems considering multiple objectives and constraints. The results obtained and the problem independent nature of the approach are encouraging for the application of the method in other multi-objective optimization problems. Future research work will be carried out studying the effect of changes in the manufacturing environment and investigating an flexible job shop scheduling model (FJSP) and heuristic solving procedures which can apply the approach proposed.

Acknowledgements

Early period (2004-2007) of the research and development was partially supported by the NODT project entitled "VITAL, Real Time Cooperative Enterprises" (National Office for Development and Technology founded by the Hungarian Government, Grant No.: 2/010/2004, project leader: László Monostori).

REFERENCES

[1] ALLAHVERDI, A., NG, C. T., CHENG, T. C. E., KOVALIOV, M. Y.: *A Survey of Scheduling Problems with Setup Times or Costs*. European Journal of Operational Research, 187, (3), pp. 985-1032, 2008.

[2] AYTUG, H., LAWLEY, M. A., MCKAY, K., MOHAN, S., UZSOY, R.: *Executing Production Schedules in the Face of Uncertainties: A Review and some Future Directions*. European Journal of Operational Research, 161, pp. 86-110, 2005.

[3] BAYKASOĞLU, A., ÖZBAKIR, L., DERELI, T.: *Multiple Dispatching Rule Based Heuristic for Multi-Objective Scheduling of Job Shops Using Tabu Search*. In Proceedings of the 5th International Conference on Managing Innovations in Manufacturing, pp. 396-402, Milwaukee, USA, 2002.

[4] KIS, T, PESCH, E.: *A Review of Exact Solution Methods for the Non-Preemptive Multiprocessor Flowshop Problem*. European Journal of Operational Research, 164, (3), pp. 573-695, 2005.

[5] KULCSÁR, GY., ERDÉLYI, F.: *A New Approach to Solve Multi-Objective Scheduling and Rescheduling Tasks*. International Journal of Computational Intelligence Research, 3, (4), pp. 343-351, 2007.

[6] LINN, R., ZHANG, W.: *Hybrid Flow Shop Scheduling: A Survey*. Computers and Industrial Engineering, 37, (1-2), pp. 57-61, 1999.

[7] LOUKIL, T., TEGHEM, J., TUYTTENS, D.: *Solving Multi-Objective Production Scheduling Problems Using Metaheuristics*. European Journal of Operational Research, 161, pp. 42-61, 2005.

[8] PATERNINA-ARBOLEDA, C. D., MONTOYA-TORRES, J. R., ACERO-DOMINGUEZ, M. J., HERRERA-HERNANDEZ, M. C.: *Scheduling Jobs on a k -stage Flexible Flow-Shop*. Annals of Operations Research, 164, (1), pp. 29-40, 2008.

[9] QUADT, D., KUHN, H.: *A Taxonomy of Flexible Flow Line Scheduling Procedures*. Journal of Operational Research, 178, pp. 686-698, 2007.

[10] SBALZARINI, L. F., MÜLLER, S., KOUMOUTSKOS, P.: *Multiobjective Optimization Using Evolutionary Algorithms*. In Center of Turbulence Research, Proceedings of the Summer Program 2000, pp. 63-74, 2000.

[11] SMITH, K. L., EVERSON, R. M., FIELDSEND, J. E.: *Dominance Measures for Multi-Objective Simulated Annealing*. In Proceedings of Congress on Evolutionary Computation, pp. 23-30, 2004.

[12] TÓTH, T., ERDÉLYI, F., KULCSÁR, GY.: *Decision Supporting of Production Planning and Control by means of Key Production Performance Measuring Indicators*. Seventh International Symposium on Tools and Methods of Competitive Engineering, TMCE 2008, April 21–25, Kusadasi, Turkey, pp. 1201-1215, 2008.

[13] VIEIRA, G., HERMANN, J., LIN, E.: *Rescheduling Manufacturing Systems: A Framework of Strategies, Policies and Methods*. Journal of Scheduling, 6 (1), pp. 35-58, 2003.

[14] WANG, W.: *Flexible Flow Shop Scheduling: Optimum, Heuristics, and Artificial Intelligence Solutions*. Expert Systems, 22, (2), pp. 78-85, 2005.

[15] YAMADA, T.: *Studies on Metaheuristics for Jobshop and Flowshop Scheduling Problems*. PhD Thesis, Kyoto University, Japan, 2003.

[16] ZHU, X., WILHELM, W. E.: *Scheduling and Lot Sizing with Sequence-Dependent Setup: A Literature Review*. IIE Transactions, 38, pp. 987-1007, 2006.