# MULTI-OBJECTIVE SEARCHING METHODS FOR SOLVING SCHEDULING AND RESCHEDULING PROBLEMS 

Gyula Kulcsár ${ }^{1}$, Ferenc Erdélyi ${ }^{2}$<br>${ }_{1}{ }_{\text {assistant professor, }}{ }^{2}$ senior researcher<br>1,2 Department of Information Engineering, University of Miskolc, Hungary<br>${ }^{2}$ Production Information Engineering Research Team (PIERT) of the Hungarian Academy of Sciences


#### Abstract

A new approach to solve multi-objective production scheduling and rescheduling problems is presented in the paper. Relationships between production goals and heuristic solving methods in an extended flexible flow shop environment are also discussed. The production goals are expressed by specifying objective functions and special production constraints are developed for the extended flowshop scheduling model. The focus is set to creating near-optimal feasible schedule considering multiple objectives. The developed methods are based on well-known searching algorithms but the applied relational operators are redefined for the multiobjective aim.


Keywords: scheduling, rescheduling, multi-objective optimization, searching algorithm,

## 1. INTRODUCTION

In a multi-objective scheduling problem class, we wish to find such a feasible schedule which optimizes a set of objective functions and subjects to a set of welldefined special constraints. The task is NP-hard therefore the "optimal" schedule is defined as a result of evolving process in which an engineer or a computer program may reach the desired (and compromised) values of the scheduling variables.

Meta-heuristics (i.e.: genetic algorithms, simulated annealing and tabu search) have become successful methods of choice for optimization problems that are too complex to be solved using deterministic techniques (see i.e. [1], [2], [5], [8], [9], [10]). To solve a multi-objective scheduling problem, it is necessary to answer an additional question: What does it mean: "good" schedule? It is not easy to specify the answer in mathematical form because of in real-life situations there are many objectives (based on delivery capability, machine utilization rate, stock level) and they are usually conflicting. The actual importance of objectives can vary frequently in time. A typical appearance of the problem in customized mass production is presented by the authors in [4].

The current paper describes a new approach that can be used in well-known meta-heuristics for comparing schedules in accordance with multiple objectives. An application of the approach is presented to the multi-objective extended flow shop scheduling and rescheduling problems.

## 2. KNOWN MULTI-OBJECTIVE OPTIMIZATION APPROACHES

Optimization problems often involve more then one aspect so it is required to use multiple criteria simultaneously. For these optimization problems the goal is:

$$
\begin{equation*}
\min _{s \in S}\left(f_{k}(s)\right) \quad \forall k \in\{1,2, \ldots, \quad K\}, \tag{1}
\end{equation*}
$$

where $s$ is a solution, $S$ is the set of feasible solutions and functions $f_{k}$ are the objectives. An objective is a measure to evaluate the quality of the solution from a given point of view.

A set of solution is said to be Pareto set if passing from solution $s_{A}$ to another solution $s_{B}$ in the set, any improvement in one of the objective functions from its current value would cause at least one of the other objective functions to deteriorate from its current value.

In the literature, different approaches can be found considering multi-objective scheduling problems, as they are surveyed i.e. in [2] and [5]. Four main approaches are as follows:

Simultaneous method (or Pareto approach) aims to generate the complete Pareto set or to approximate a set of efficient solutions.

Weighting objectives method creates a weighted linear combination of the objectives to obtain a single function, which can be solved using any single optimization method.

Hierarchical optimization method allows the decision maker to rank the objectives in a descending order of importance, from 1 to $K$. Each objective function is then minimized individually subject to a constraint that does not allow the minimum for the new function to exceed a prescribed fraction of a minimum of the previous function.

Goal programming method takes the objectives into constraints which express satisfying goals. The aim is to find a solution which provides good values of predefined goals for each objective.

## 3. A NEW METHOD BASED ON RELATIONAL OPERATORS

Our new idea is that the relative goodness of a solution is more important than the absolute goodness of one. The base of the method is that we measure the relative goodness of the selected solution by comparing it with another solution in the feasible region.

Let $S$ be the search space under consideration. That is, it is the set of all possible solutions to our problem. Suppose that we have a number of objective functions $f_{1}, \ldots, f_{K}$ such that:

$$
\begin{equation*}
f_{k}: S \rightarrow \mathfrak{R}, \forall k \in\{1,2, \ldots, K\} . \tag{2}
\end{equation*}
$$

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 many cases, it will not be possible to find 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 among the various criteria used to evaluate the quality of solutions.

More formally, let $s_{x}, s_{y} \in S$ be two solutions. We define the function $F$ :

$$
\begin{equation*}
F: S^{2} \rightarrow \mathfrak{R}, F\left(s_{x}, s_{y}\right)=\sum_{k=1}^{K}\left(w_{k} * D\left(f_{k}\left(s_{x}\right), f_{k}\left(s_{y}\right)\right)\right) \tag{3}
\end{equation*}
$$

in which $D$ means the following function:

$$
D: \mathfrak{R}^{2} \rightarrow \mathfrak{R}, a, b \in \mathfrak{R}, D(a, b)=\left\{\begin{array}{l}
0, \text { if } \max (a, b)=0  \tag{4}\\
\frac{b-a}{\max (a, b)} * 100, \text { otherwise }
\end{array} .\right.
$$

The max denotes an operator:

$$
\max : \mathfrak{R}^{2} \rightarrow \mathfrak{R}, \max (a, b)=\left\{\begin{array}{l}
a, \text { if } a>b  \tag{5}\\
b, \text { otherwise }
\end{array}\right.
$$

Moreover, to express the importance of any component objective function $f_{k}$, we use $w_{k}$ weighting coefficient which is an integer value within range [ $0, \ldots, 10$ ]. It is allowed decision maker to set the actual weights of each objective function.

Function $F$ is characterized by anti-symmetry:

$$
\begin{equation*}
F\left(s_{x}, s_{y}\right)=-F\left(s_{y}, s_{x}\right), \tag{6}
\end{equation*}
$$

and transitivity:

$$
\begin{equation*}
\text { if } F\left(s_{x}, s_{y}\right)<0 \text { and } F\left(s_{y}, s_{z}\right)<0 \text { than } F\left(s_{x}, s_{z}\right)<0, s_{x}, s_{y}, s_{z} \in S . \tag{7}
\end{equation*}
$$

Using these features of the function $F$, we tell about two solutions $s_{x}, s_{y} \in S$ that: $s_{x}$ is better solution than $s_{y}\left(s_{x}<s_{y}\right.$ is true) if (8) is true.

$$
\begin{equation*}
F\left(s_{x}, s_{y}\right)>0 . \tag{8}
\end{equation*}
$$

$s_{x}$ and $s_{y}$ are equal ( $s_{x}=s_{y}$ is true) if (9) is true.

$$
\begin{equation*}
F\left(s_{x}, s_{y}\right)=0 . \tag{9}
\end{equation*}
$$

$s_{x}$ is worse solution than $s_{y}\left(s_{x}>s_{y}\right.$ is true) if (10) is true.

$$
\begin{equation*}
F\left(s_{x}, s_{y}\right)<0 . \tag{10}
\end{equation*}
$$

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

## 4. AN APPLICATION TO EFFS SCHEDULING PROBLEM

### 4.1. Problem description

In the extended flexible flow shop (EFFS) scheduling model, there are different final products which may be produced. There are an order book for a given time period. It has production orders. Each production order includes the identification of the final product, the required quantity and the demanded due date. At the shop floor, product-pallets can be moved between machines. Each pallet consists of a pre-decided number of the finished products. Each production order is identified to be consisting of a particular number of pallets. We schedule pallets; one pallet means one job. Each job has four attributes: 1. the type of the final product, 2. the quantity of the products, 3 . the start (release) time (the earliest time when all of the required material available in the needed quantity) and 4. the demanded due date.

Each job has to visit four technology steps in the same sequence. A technology step may include some operations, but no pre-emption is allowed at the level of the technology steps.

The workshop contains different machine groups connected to each others in a given configuration. Each machine group contains a pre-defined number of machines. In a given machine group, each machine can process the same execution step which is a well-defined set and sequence of technology steps.

The machines are not continually available for processing, therefore they have one or more non-availability intervals. In addition, each machine may have different production rates (quantities producible per time unit) for different products. Similarly, each machine may be characterized by product sequence dependent setup times (time delay to changeover from one product type to another product type).

A given final product can be produced differently, because there are different execution routes on which the required components are taken through becoming the final product.

The shop floor has already been loaded, the actual state of the system is known. It means that the effect of the last confirmed schedule must be obtained.

### 4.2. Objective functions

For due date related objectives, we assume that there are production orders (PO), jobs $J_{i}(i=1, \ldots, N)$ and manufacturing tasks $O_{t}(t=1, \ldots, Z)$. Each job $i$ has due date $d_{i}$. The completion time of job $i$ is denoted by $C_{i}$. The lateness of the job $i$ is:

$$
\begin{equation*}
L_{i}=C_{i}-d_{i}, \tag{11}
\end{equation*}
$$

and the tardiness of the job $i$ is:

$$
\begin{equation*}
T_{i}=\max \left(0, L_{i}\right) . \tag{12}
\end{equation*}
$$

In current study, five objective functions are considered for demonstrating multi-objective predictive scheduling. These are as follows: (13) the number of tardy jobs, (14) the sum of tardiness, (15) the maximum tardiness, (16) the number of setups, (17) the maximum completion time (makespan).

$$
\begin{gather*}
f_{1}=\left\{\left\{J_{i} \mid T_{i}>0\right\} \mid\right.  \tag{13}\\
f_{2}=\sum_{i} T_{i}  \tag{14}\\
f_{3}=\max _{i}\left(T_{i}\right)  \tag{15}\\
f_{4}=\left\{O_{t} \mid \text { setuptime }_{t}>0\right\} \mid  \tag{16}\\
f_{5}=\max _{i}\left(C_{i}\right) \tag{17}
\end{gather*}
$$

### 4.3. Heuristic method for creating near-optimal schedule

For solving the problem addressed above, we use a tabu search algorithm based on simultaneous production simulation, overloaded relational operators and neighboring operators.

Tabu search is a special search procedure which iteratively moves from a schedule $s_{x}$ to a schedule $s_{x}{ }^{\prime}$ in the neighborhood of $s_{x}$, until some stopping criterion has been satisfied. To explore regions of the search space that would be left
unexplored by a simple local search procedure and escape local optimality, tabu search modifies the neighborhood structure of each schedule as the search progresses. A tabu list contains the schedules that have been visited in the recent past (less than a given number of moves ago). Schedules in the tabu list are excluded from the neighborhood of the actual solution.

A certain number of neighbors of the current schedule are generated randomly by neighboring operators. These operators create new feasible schedules by modifying resource allocations and job sequences. The objective functions concerning schedules are evaluated by a production simulator which represents the machine environment with capacity and technological constraints. The simulation means numerical simulation of the production to calculate the performance of the given schedule. The method proposed in section 2 is used to compare the generated schedules according to multiple objectives described in section 4.2. The best schedule becomes the initial solution of the next step.

When the scheduling process is finished or stopped by user, the currently best known schedule is returned, so the method can be used in any-time working model.

Basing on the results of tests which are executed on sample problem instances, we can say that the proposed method is able to find good solutions to the problem in a reasonable amount of time.

## 5. RESCHEDULING PROBLEMS

In the following, we shortly deal with the practical issues of predictive-reactive rescheduling task. In practice, generating high quality predictive schedule according to multi-objectives is not enough because many unexpected events require the revision and modification of the released schedule. 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. Different type of uncertainty can occur i.e. 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. Time shift rescheduling postpones executions of certain tasks and jobs in time, but their resource assignments and sequences are not changed. Partial rescheduling modifies only jobs and resources affected by the disruption. Complete rescheduling generates a new feasible schedule. These methods are presented in detail in [11].

It is required of rescheduling methods to consider new demands added to 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. More research is needed to define qualitative indices (i.e. related to setup and due date) for supporting comparison of schedule changes. Such performance indices can be considered as objective functions of rescheduling and they can be used by multiobjective scheduling methods proposed in the paper.

## 6. CONCLUSIONS

This paper described the proposition and application of a new method for multiobjective scheduling and rescheduling problems. It is based on new interpretation and usage of relational operators for schedules in searching algorithms. After developing computer program, the concept is successfully tested on extended flexible flow shop scheduling and rescheduling problems (originated from an industrial project) under multiple objectives. The obtained results and the problem independent nature of approach are encouraging to apply the method in other multiobjective optimization problems.

## ACKNOWLEDGEMENTS

The research and development summarized in this paper was partially supported by the Hungarian Academy of Sciences (HAS) within the framework of Production Information Engineering Research Team (PIERT) established at the Department of Information Engineering of the University of Miskolc (Grant No. MTA-TKI 06108). The results are also connected with the NODT project entitled "VITAL" (National Office for Development and Technology founded by the Hungarian Government, Grant No.: 2/010/2004, project leader: László Monostori DSc).

## REFERENCES

[1] BAYKASOĞLU, A., ÖZBAKIR, L., DERELI, T.: Multiple Dispatching Rule Based Heuristic for Multi-Objective Scheduling of Job Shops Using Tabu Search. Proceedings of the 5th International Conference on Managing Innovations in Manufacturing, Milwaukee, USA, 2002. pp. 396-402.
[2] GEIGER, M. J.: Foundations of the Pareto Iterated Local Search Metaheuristic. Proceedings of the 18th International Conference on Multiple Criteria Decision Making, Chania, Greece, 2006.
[3] KIS, T.: Automatic Scheduling System with Basic Scheduling Engine. For GE lighting. Specification. V. 06. 2005. VITAL project Cluster 1.
[4] KULCSÁR, GY., ERDÉLYI, F.: Production Goals and Heuristic Methods in Extended Flexible Flow Shop Scheduling Problems. Forum of PhD Students, Miskolc, Hungary, 2005. pp. 104-109.
[5] KULCSÁR, GY., HORNYÁK, O., ERDÉLYI, F.: Shop Floor Decision Supporting and MES Functions in Customized Mass Production. Conference on Manufacturing Systems Development Industry Expectations, Wroclaw, Poland, 2005. pp. 138-152.
[6] LOUKIL, T., TEGHEM, J., TUYTTENS, D.: Solving Multi-Objective Production Scheduling Problems Using Metaheuristics. European Journal of Operational Research, 161, 2005. pp. 42-61.
[7] MCKAY, K. N., WIERS, V. C.: Unifying the Theory and Practice of Production Scheduling. Journal of Manufacturing Systems, Vol. 18, No. 4, 1999. pp. 241-248.
[8] SBALZARINI, I.. F., MÜLLER, S., KOUMOUTSAKOS, P.: Multiobjective Optimization Using Evolutionary Algorithms. Center of Turbulence Research, Proceedings of the Summer Program 2000, 2000. pp. 63-74.
[9] SMITH, K. I.., EVERSON, R. M., FIELDSEND, J. E.: Dominance Measures for Multi-Objective Simulated Annealing. Proceedings of Congress on Evolutionary Computation, 2004. pp. 23-30.
[10] YAMADA, T.: Studies on Metaheuristics for Jobshop and Flowshop Scheduling Problems. Ph.D Thesis, Kyoto University, Japan, 2003.
[11] VIEIRA, G., HERMANN, J., LIN, E.: Rescheduling Manufacturing Systams: A Framework of Strategies, Policies and Methods. Journal of Scheduling, Vol. 6, No. 1, 2003. pp. 35-58.

