MANUFACTURING SYSTEMS SCHEDULING THROUGH MACHINE LEARNING

Paolo Priore, David de la Fuente García, Isabel Fernández Quesada

ETSII e II, CAMPUS DE VIESQUES, 33204 GIJON

Telf.: (98)5182107 e-mail: priore@etsiig.uniovi.es

The problem of manufacturing systems scheduling by means of dispatching rules is that these rules depend on the state in which the system is in every moment. Therefore it would be interesting to use in every state of the system, the most adequate dispatching rule to that state. To achieve this goal, it is presented in this paper a scheduling approach which uses machine learning. This approach, according to the previous performance of the system (training examples), is capable of working out the most appropriate dispatching rule for every state of the system.

**Keywords:** Scheduling, manufacturing systems, machine learning

1. Introduction

Scheduling is part of the operational control process in a manufacturing system. When a common set of resources in the manufacturing system must be shared, scheduling is needed to make a variety of different products during the same period of time. The aim of manufacturing scheduling is the efficient allocation of machines and other resources to jobs, or operations within jobs, and the subsequent time-phasing of these jobs on individual machines (Shaw et al.[6]).
According to Mayer, Phillips, and Young [3] over 98% of the time work-in-process is involved in non-value-added activities. Proper scheduling of activities is an important tool for improving this situation. However, scheduling is rather a difficult task since manufacturing environments have high levels of inaccuracy, the requirements for processes are detailed and specific, and management goals are varied, dynamics and often conflicting.

Most traditional approaches to solving scheduling problems use either simulation analysis, analytical models or combinations of these methods. Simulation appears to be the appropriate tool to analyse the effect of different releasing and dispatching policies (see, for example, Baker [2] and Stecke et al. [7]). The difficulty of the analytical approaches is the complexity of the scheduling problem as even the most simplified models for scheduling are NP hard problems which results in the impossibility of optimally solving realistically sized problems.

Over the last fifteen years, an important effort has been dedicated to developing Artificial Intelligence (AI) methods for scheduling. AI methods are often called knowledge-based systems. Machine learning lays in the field of AI.

Machine Learning is a rapidly growing research area for studying methods for developing AI systems that are capable of learning (Michalski [4]). The ability to learn and improve is essential for intelligent systems; however, little work has been done in applying machine learning to intelligent scheduling (Shaw et al. [6]). Aytug et al. [1] display a review of Machine Learning in scheduling.

2. Machine Learning
From a computational viewpoint, the problem of learning is to create computable procedures to perform tasks of which only partial descriptions about the way the system is desired to work are known. In very general terms the machine learning scheme is depicted in figure 1.1.

A particular case of machine learning is when the poorly structured information source of which the machine learning algorithms make use is given by a collection of examples which can be considered as a sample of the performance of that which is wanted to be learned.

```
Fig. 1.1. Machine learning general scheme
```

Particularly, this examples, which are usually called training examples, will be described by attribute-value tables, where there will be a special attribute known as the class. The aim is to try to learn to classify new cases similar to those in the examples, but about which only the values of the different attributes are known but not that of the class. Machine learning can be considered as a problem of creation in which knowledge is built. There is a great variety of knowledge representations, but the most used are the rules like the following:
IF \((b_{i1} \geq a_i \geq c_{il})\) \(\ AND \ \ \ldots (b_{in} \geq a_i \geq c_{in})\) THEN \(C_i\)

where \(a_i\) represents the \(j\)-th attribute, \(b_{ij}\) and \(c_{ij}\) define the range for \(a_i\), and \(C_i\) denotes the class and the trees of decision which main feature is the fact that the branches are used to represent questions about the values of the attributes and include in the leaves the decisions about the classes to which presumably the object to be classified will belong.

3. Machine Learning-based Scheduling

The suggested approach consists of five basic steps, as it is shown in figure 3.1. The steps are the following:

1. Definition of control attributes for capturing the relevant manufacturing patterns.
   Obviously it is not possible to take into account all possible attributes. A synthesis must be done.

2. Creation of training examples using different values of the control attributes. These values must be the common ones in the manufacturing system studied. For every training example, the rule of dispatching with the best performance, must be worked out through simulation. This rule will be the class for this training example.

3. Achieving heuristic rules by means of a learning program. One heuristic rule should look like the next one:

\[
IF \quad (b_{i1} \geq a_i \geq c_{il}) \quad AND \quad \ldots (b_{in} \geq a_i \geq c_{in}) \quad THEN \quad \text{rule dispatching}
\]

4. Using the previously calculated heuristic rules to select the most adequate dispatching rule depending on the set of values of the control attributes present in every moment.
5. To compare the performance of the heuristic rules versus the single dispatching rule which best performs in every moment. Should the first rules perform worse than the single dispatching rule, more training examples must be generated.

4. Experimental Study

The system considered in this paper, is a simplified two work-centre flow shop: WC1 and WC2. Each work centre has two machines capable of doing the same operations. The set of dispatching rules used is the following:

1. SPT. The operations are ordered according to the shortest operation time first.
2. LPT. The operations are scored according to the longest operation time first.
3. EDD. The operations are ordered according to the earliest due date first.
4. FCFS. The operations are scored according to the rule first come first served

The control attributes selected for generating the training examples are the following:

1. Mean arrival rate of jobs
2. Expected processing times on WC1
3. Expected processing times on WC2
4. Flow allowance factor (F) which measures due date tightness. Following Baker [2], the Total Work Content (TWK) rule for assigning job due dates has been employed. Under TWK, the due date \( d_i \) of job \( i \) is worked out as follows:

\[
    d_i = a_i + F \times p_i
\]

where \( a_i \) is the arrival time, and \( p_i \) is the processing time of the job \( i \).

The system performance criteria used in this paper is the mean tardiness, being this criteria the most used in all manufacturing systems.

In the simulation experiments during the learning stage, it was supposed that the jobs arrived according to a Poisson distribution. The processing times followed an Exponential distribution and the Flow Allowance Factor varied within 2 to 4. Job interarrival times were varied to result in an overall system utilization within 60% to 95%.

The experiments addressed 400 different combinations of these parameters. In each case, the steady state mean tardiness values resulting from employing the dispatching rules singularly were determined and the best one was chosen as the class for that combination.

To obtain the heuristic rules it was used the learning program C4.5 (Quinlan [5]). The learning process in C4.5 follows a sequence of specialization steps guided by an information entropy function for evaluating class membership (Shaw et al. [6]). The number of heuristic rules obtained was 7. The EDD rule is chosen in 4 out of the 7 and the SPT one in the 3 left. The default class is SPT.

Afterwards, various test simulations were generated by using new combinations of the parameters used in the training process and different number of changes in this combinations for each simulation. It was observed that in general terms the heuristic rules performed better than
the best dispatching rules in each case. The number of test simulations were 30. The heuristic rules were better in 14 out of these 30, in 11 were as good and in 5 they were worse. The mean performance improvement was 8.5%.

5. Conclusion

In this paper, we have proposed a scheduling approach which employs learning techniques. In general terms, the performance with this approach is better than the one using a dispatching rule individually. One of the inconveniences of this approach is the fact that a great deal of computing time is required for the simulations. This simulations, however, only have to be executed once. This paper is just the first step. We intend to keep on studying this approach in more complex manufacturing systems as Job Shops or FMSs, as well as this approach performance under system disturbances as machine breakdowns or rushed jobs.

6. References


