Generation of knowledge about the control of a flow shop using data-analysis oriented learning techniques and simulation

Henri Pierreval, Henri Ralambondrainy

UNITE DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 897

GENERATION OF KNOWLEDGE ABOUT THE CONTROL OF A FLOW SHOP USING DATA-ANALYSIS ORIENTED LEARNING TECHNIQUES AND SIMULATION

Programme 5

Henri PIERREVAL
Henri RALAMBONDRAINY

Septembre 1988

GENERATION OF KNOWLEDGE ABOUT THE CONTROL OF A FLOW SHOP USING DATA-ANALYSIS ORIENTED LEARNING TECHNIQUES AND SIMULATION

GENERATION DE CONNAISSANCES POUR LE CONTROLE D'UN ATELIER DE TYPE FLOW-SHOP UTILISANT DES TECHNIQUES D'APPRENTISSAGE ET DE SIMULATION

Henri PIERREVAL (1), Henri RALAMBONDRAINY (2)

(1) CIRILLE : Laboratoire d'informatique et de modélisation des systèmes de production. Université Claude Bernard LYON I, Bat. 710 43, Bd. du 11 novembre, 69622 VILLEURBANNE Cedex, FRANCE,

(2) INRIA : Domaine de Voluceau Rocquencourt B. P. 105 78153 LE CHESNAY Cedex, FRANCE.

RESUME :

Une des difficultés essentielles lors du développement de systèmes experts d'aide à la conduite d'atelier (ou d'ordonnancement), est de trouver la connaissance nécessaire. Cet article présente sur l'exemple d'un flow shop simplifié, l'utilisation d'un algorithme d'apprentissage, permettant de générer, à partir des résultats de simulations, des règles de production directement exploitables par un système expert.

ABSTRACT :

One of the most important problem which occurs when developing expert systems in manufacturing control (or scheduling), is to find the required knowledge. Learning algorithm, are able to generate, from simulation experiments, a set of production rules which can be inserted in an expert system knowledge base. This approach is illustrated with GENREG, on the case of a simplified flow shop.

KEY WORDS : Manufacturing systems, learning, simulation, expert system, manufacturing control, scheduling

1. INTRODUCTION

Usually, production management problems are decomposed into different hierarchical decision levels. In this paper, we focus on the lower levels, that we globally call, for the sake of simplicity, manufacturing control (the approach may include schedulingand sequencing). The manufacturing control is in charge of managing the flow of the products and resources, by making decisions in real time (see (Pierreval 87a) for a taxonomy of these decisions). Often these decisions are particularly difficult to make, because of the complexity of most manufacturing systems. Moreover the workshop control must deal in the better way with : production objectives (reducing work in process, ...), directives from higher levels (production planning...), several disturbances (breakdowns, urgent orders, absenteeism...).

Expert systems (ES) are proposed in several works as a fertile technique to cope with these problems. Unfortunately, in most cases, the development of their rules base is difficult, because of the lack of required knowledge.

A new approach as been introduced in (Pierreval 88), based on the application of learning techniques to simulation experiments. In this paper the example of a simplified flow shop problem is presented, in order to demonstrate the principles and the interesting capabilities of the approach.

2. EXPERT SYSTEMS IN MANUFACTURING CONTROL

Due to the limits of mathematical-oriented approaches in dealing with manufacturing control and scheduling problems, the use of AI-oriented techniques, is studied by many researchers. Several expert system approaches, related to different manufacturing problems are proposed. They are based on :

- fuzzy production rules ,
- PROLOG ,
- LISP...

Using ES in manufacturing control presents several advantages (Bel 85) :

- to describe in the same programming environment the strategies of production managers or specialized operators, and those proposed by more theoretical works,

- strategies can be updated during the evolutions of the production objectives, or when the work shop configuration is modified,

- software transportations from one shop to an other are easier (the chosen knowledge representation reduces the required modifications).

Nevertheless, building such expert systems is a difficult task. It is known that the required knowledge is difficult to obtain, considering the complexity of manufacturing systems. Even people working in the factory have not all the knowledge required to develop a rule base.

3. CONTRIBUTION OF SIMULATION

Simulation has been widely used to evaluate and compare dispatching or sequencing rules (Hershauer 75), (Barett 86), or more complex control strategies (Pierreval 87b). In many cases, it has been applied to a given type of system (flow shop, job shop,...). Only a few works deal with the variation of a sufficient number of parameters describing the system and the production (arrival rate of parts, mean operating time...). Unfortunately the contribution of these works is limited. First because it is impossible to extrapolate the results to other systems with differents characteristics (more or fewer machines, different production plans, rework necessity, breakdowns,...), second because the inclusion of many parameters in the simulation, which give rise to various tables of results, which are difficult to read, and hard to interpret.

It has been stated that simulation can contribute to the development of an ES rules base : by connecting the ES to a simulation model of the shop, it is possible to test rules and to evaluate their contribution to the system performances (Bel 85), (Pierreval 86). Moreover, the advantages of modeling of the decisional component of certain workshop with an ES for their simulation are demonstrated in (Pierreval 87b), through the example of a foundry simulation. Furthermore interesting works have been done, which show the benefits of using applying techniques based on classification and clustering to simulation samples (Bonneau 85), (Cannals 86).

According to these points, we are interested in using simulation to find rules required for building knowledge bases, or in improving existing ones. The proposed approach consists in evaluating the performances of different possible control strategies, on a wide range of system configurations, using simulation. This provides a sample of "observations" of the system behavior when it is controlled in different manners. It is possible to learn from these observations thanks to specialized learning methods.

4. INDUCTION OF PRODUCTION RULES

4.1 Learning techniques :

In recent years, different kind of learning techniques have been developed (Learning by Discovery, by Analogy, from Examples, ... see (Michalski 84) for references), which are able to generate new knowledge through observations and experimentations. We are interested in this paper with techniques so-called Learning from Examples. From a given concept, described by a set of examples and counter-examples, these kind of techniques research, in a space of possible descriptions, a set of rules which recognize the examples and not the counter-examples. One of well known learning system is the ID3 program (Quinlan 1979) which from features vectors, build a tree structure for decision rules. The AQ11 and INDUCE (Michalski 1984) use predicate calculus notation to represent both data and the induction researched rules. They were sucessfully applied to a problem of soybean diseases.

In our case, a table of simulation results may be considered as a training set related to different states of the system behavior. The concept to be recognize will be here a particular behavior of the system. For instance, we can characterize the system state "good performance" with good scores on some special criteria. The goal of a learning system will be to find what strategies, according to a given configuration, will be the best suited to move the system in this state.

4.2 The induction algorithm GENREG :

The observations in input must be represented by a set of couples (attribute, value). The attributes or descriptors may have a hierarchy or graph structure (background-knowledge). The program looks for a set of assertions which recognize the examples and not counter-examples. More precisely the decision rules must verify the following properties :

-each assertion must not recognize no more than a given a priori number of counter-examples α. This condition allows to recognize concept which are imperfectly characterized by the set of examples and counter-examples.

-each assertion recognizes at least a given a priori number of examples β. This condition guaranties the robustness of the rules and eliminates too particular rules.

The quality of an assertion is measured by the ratio : number of examples recognized/number of observations recognized.

Two algorithms are provided. The first one is a *data driven* type. It starts from the observations, and generalize them until it is not possible to find new assertions which verify the two previous conditions (If A and B are two assertions, B is said more general than A, if A $\Rightarrow$ B). The second one is a *model driven* type, it starts from general assertions and specialize them to find good assertions according to the previous conditions. For more details see (Ralambondrainy 1988).

5.GENERAL PROBLEM FORMULATION (Pierreval 88).

5.1.System configuration

Let X=(X1, X2,..., Xp) be a vector of descriptive variables of the system, called the system configuration. Each Xi takes its values in a set Ei (often an interval of R). The Xi variables are those that must be taken into account for manufacturing control, in particular :

- characteristics of the workshop during a given period, for example : number of machines or operators available, mean expected time of breakdown, ...,

- characteristics of the production plan during a given period,for example : the mean arrival rate of jobs, mean expected processing time on work stations, number of different part references ...

- the state of certain system variables at a given instant, for example the status of machines (breakdown, free, ..), the status of certain operators, the number of parts in a given buffer...Several Xi may represent the same system variable at different instants.

5.2 Control strategies

Let S = S1, S2, ..., Ss be a set of control strategies, which may be :

- specific to a given situation, such as what to do when a bottleneck appears at a workstation, or how to allocate jobs when a given resource becomes unavailable ... ,

- general strategies that can be used on given production periods, according to given objectives. Well known examples are dispatching rules (and sequencing rules), that are widely presented in several studies (Hershauer 75)...

A dispatching rule is used to select the next job to be processed from a set of jobs awaiting service. These rules are either static or dynamic (state dependant). The most well known are SPT (Shortest Processing Time served first), LPT (Longest Processing Time served first), EDT (Earliest Due Date served first), FIQ (First In Queue served first), FAS (First Arrived in System served first), ...

5.3 Performance criteria

Let C = (C1, C2, ..., Cm) be a vector of performance criteria that are of interest as regards to the production objectives. Each Ci takes its values in Oi (assumed to be an interval of R). Examples of Ci are the mean and standard deviation of : flowtime (amount of time the jobs spend in the systems), lateness (amount of time by which the completion time of jobs exceed their due dates), tardiness (the positive lateness of jobs)...These criteria may also be specific to certain situations, for example the amount of time spent for a bottleneck resorption...

Generally the goal of the production objectives is to minimize (sometimes to maximize) the performance criteria. Unfortunately in many cases, there is not a strategy that is globally more efficient than the others : certain Si give interesting results as regards to certain Cj, according to certain x from X. Moreover, some Ci may be antagonistic.

Our purpose is to be able to formalize knowledge about the system behavior. For example to find rules, that allows the selection of a strategy, that is the best suited to a given x, according to a given production objective (function of C).

6. SIMULATION OF A SIMPLIFIED FLOW SHOP

In the following, we illustrate our approach on the example of a scheduling problem, solved in real time thanks to dispatching rules. This requires a study based on steady state simulation of the system. Meanwhile, the approach may also be applied to studies starting from a given initial state at a given instant (Pierreval 88).

This example is based on an article from R. T. Barett, and S.Barman ; more details about the system and the assumptions for the simulation can be found in (Barett 86). The system is a simplified two-work centers flow shop : WC1 and WC2. Each work center has two machines capable of doing the same operations. Some jobs need reworks, done at WC2. The aim of the original study was to evaluate the influence of dispatching rules at each work center.

A simulation model of the flow-shop introduced as example 1 has been developed using SIMAN (Pegden 85), in order to take into account more parameters on the system configuration than in the original study.

Si is decomposed in strategies for WC1, and strategies for WC2. There are :

S-WC1 : FIQ, EDT, SPT, LPT.
S-WC2 : FIQ, FAS, EDT, SPT, LPT.

Their evaluation must be done according to the system performances. Different variables characterized here, only by the mean were used :

C1 : flowtime,
C2 : lateness,
C3 : waiting time,
C4 : tardiness,
C5 : earliness,
C6 : jobs in process.

The system configuration is characterized by the following variables. They were broken into classes with the same probability (the method will be described further) :

X1 : mean arrival rate of jobs (r1, r2, r3),

X2 : expected processing times on WC1 (three classes : p11, p12,p13),
X3 : expected processing times on WC2 (p21, p22, p23),
X4 : processing time variation (v1,v2 : the 2 initial values).

7. DESCRIPTION OF THE SIMULATION EXPERIMENTS

The simulation study must take into account various system configurations, i. e. various x from X. Since the objective is to analyze the influence of the Si on C, experiments must be conducted in such a way that the choice of the x values does not give rise to bias. In the example X and S must be independent, the chosen x values must constitute a representative sample of the system characteristics, and the Xi of X must independent. In order to take into account these constraints, the experiments were conducted as follows:

7.1 Preliminary steps

We must define each Ei, so that the system can reach a steady state. This can be done with pilot simulation runs, using the strategies which are thought to be the least performant. We used LPT on each work center, to define the minimum and maximum of : the mean arrival rate of jobs, the mean expected times on each work centers. To do this, deferent high values for each variables, were simulated. The existence of a steady state, for each set of values tested, was verified thanks to bit mapped graphic barcharts of the variables works in process and time in system, using the SIMAN OUTPUT processor. The lowest values were defined so that the works in process were not too low.

Truncations in order to avoid the initial bias, and the run durations (that must be sufficiently long), can be defined in these preliminary steps. For the example, we kept the same values than the initial study.

7.2 The experiments

The strategies si must be evaluated with many system configuration x. To do this the xi are randomly selected in Ei. The N (=198 in our case) selected values of x=(x1, ...,x4) constitute a sample of system configurations. Each x is used as simulation input for comparing the Si.

Given x, 19 simulations (cartesian product of S-WC1 and S-WC2 except LPT-LPT), must be performed, in order to evaluate the performance of each Si by collecting and estimating 19 values of c=(c1, ..., c6) (note that FAS = FIQ for WC1).

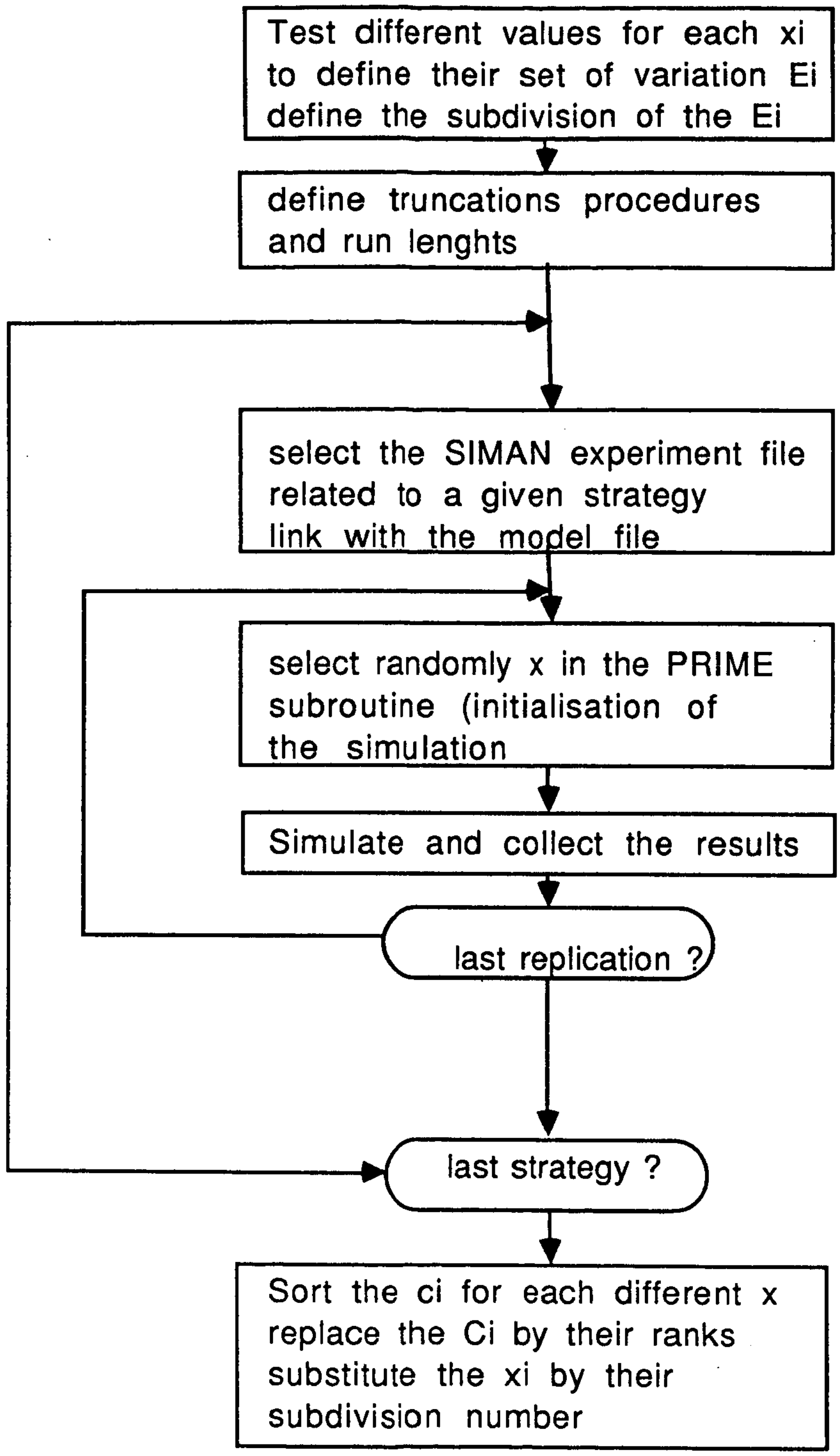
On a practical way, 19 SIMAN experiment files were used for each couple of strategies. For each of the 19 simulation runs, 198 replications allow the selection of several x in the subroutine PRIME. It is important to use carefully the random number generators, so as to use the same sample of system configuration to test the set of strategies.

For each replication, the values xi, Si, and ci are kept in a result file file.

7.3 Treatment of the result file

The results file must be homogenized before to be treated by GENREG in a proper way. Each ci value must be replaced by its rank in the set of all the observations related to a given x. This allows to avoid the direct influence of the system configuration on the performances (avoid rules such as : if the arrival rates of job is high, and if the expected processing times are high and if the processing time variation is high then the flow time is bad).

The method is summarized through the following figure.



DIFFERENT STEPS FOR THE SIMULATION EXPERIMENTS

8. RESULTS

As a first step, a multiple correspondence analysis was made on the simulation results, in order to have an overview of the interrelations between the variables (Pierrevall 88).

The simulation data were prepared as previously described, and treated by GENREG. After a preliminary study, in order to test the approach, we choose the the "best mean tardiness" as the concept to recognize. For each of the 198 different system configurations, one of the 19 couples of strategies obtains the "best" results for this variable (rank 1), i. e. the lower value on the criteria mean tardiness (in spite of the stocks which are created in case of too much earliness).

In order to be able to obtain a rules set as complete as possible, we set the GENREG parameters so that each assertion must not recognize no more than 20 counter-examples (rank greater than 1 on the criteria mean tardiness), and each assertion recognizes at least 2 examples (best rank on mean tardiness).

GENREG gave 30 rules, which recognize 100 % (198/198) of examples, and 9 % of counter examples (305/3564). Its clear that this whole performance is interesting as regards to :

- the simplicity and limited number of descriptors used (X and S),

- the statement that strategies giving the rank 2 are classified in counter examples.

The 8 rules with the greatest certainty are all based on the following strategy :

- SPT on WC1,
- EDT on WC2.

Examples of these rules are :

RULE 1 (CERTAINTY =0.8) :

IF S-WC1 = SPT
AND S-WC2 = EDT
AND EXP. PROC. TIME ON WC1 = HIGH

THEN MEAN TARDINESS = BEST

RULE 6 (CERTAINTY = 0.64) :

IF S-WC1 = SPT
AND S-WC2 = EDT
AND EXP. PROC. TIME ON WC1 = MEDIUM
AND PROC. TIME VARIATION = HIGH
THEN MEAN TARDINESS = BEST.

Other strategies appear later with other system configurations. However, their certainty is low, and these rules must be used carefully. Examples are :

RULE 10 (CERTAINTY = 0.35) :

IF S-WC1 = SPT
AND S-WC2 = SPT
AND ARRIVAL RATE OF JOBS = HIGH
AND EXP. PROC. TIME ON WC1 = MEDIUM
THEN MEAN TARDINESS = BEST,

RULE 20 (CERTAINTY = 0.22) :

IF S-WC1 = EDT
AND S-WC2 = SPT
AND ARRIVAL RATE OF JOBS = HIGH
AND EXP. PROC. TIME ON WC1 = MEDIUM
THEN MEAN TARDINESS = BEST.

Rules 10 and 20 give different strategies for the same system configuration. This allows a possible choice regarding an other performance criteria (flowtime, or standart deviation of tardiness ...) . We noted that the formula (conditional probability), used to compute the certainty, presents pessimistic values (under-estimation of the rule quality), for our case at least.

9. CONCLUDING REMARKS

In this paper, we have proposed and illustrated the application of learning techniques to simulation results. This approach provide useful knowledge for ES in manufacturing control or scheduling, in spite of the computing times required for simulations. It is clear that either strong or weak rules may be find, depending to the case. Meanwhile, it appears as a fertile technique whenever there is a lack of knowledge about a manufacturing process. Further publications will complete certain disregarded aspects of this paper.

REFERENCES

BARRETT R. T. , BARMAN S.
"A SLAM II Simulation study of a simplified flow shop",Simulation 47 : 5, Nov. 1986, p 181-189.

BEL G., BERRADA M., CORREGE M., DUBOIS D.
"SIMULISP : un banc d'essai de systèmes experts pour la commande des ateliers de production", AFCET congress "des outils pour demain", Toulouse, France, octobre1985.

BEL G, DUBOIS D, F ARRENY H., PRADE H.
"Towards the use of fuzzy rules-based systems in the monitoring of manufacturing systems", Conf. PROLAMAT, Paris, juin 1985, p.109-119.

BONNEAU F., PROTH J. M.
"Application de règles de gestion à un système de fabrication : classification des objectifs atteints en vue de leur utilisation", rapport INRIA n° 372, Mars 1985.

CANALS D.
Ordonnancement d'atelier par simulation : Etude des règles depriorité et aide au lancement,thèse de Docteur Ingénieur, ENSAE, Toulouse,juillet. 1986

HERSHAUER J. C. , EBERT R. J.
"Search and simulation selection of a Jobshop sequencing rules", Management science, Vol 21, N3 7,mars 1975, p. 833-843.

MICHALSKI.R.,CARBONELL J., MITECHELL.T.
Machine Learning Tioga Press, Palto Alto 1984

PEGDEN C. D.
Introd.to SIMAN, Systems Modeling Corp., 1985.

PIERREVAL H., DUSSAUCHOY A.
"Use of expert systems to complete traditional approaces for production systems simulation modeling", 2nd European Simulation Congress, Anvers, Belgique, septembre 1986, P. 377-383.

PIERREVAL H., DUSSAUCHOY A.
"Quelques moyens de modélisation des activités de décision entemps réel en simulation", proc. of the "IASTED Conf. Identification, Modeling, and Simulation", Paris, juin 1987.

PIERREVAL H.
Modélisation et simulation de processus industriels, Etude d'un cas ayant utilisée un langage de simulation et un système base sur des règles de production", Proc. conf. INRIA : systèmes de production, Paris, avril 1987.

PIERREVAL H.
"Data analysis oriented techniques for learning about manufacturing control with simulation", 2nd European Multiconference : factory of the future, Nice, juin 1988, p 61-66.

QUINLAN.J.
Semi-autonomous Acquisition of Pattern-based knowledge, in Introductory readings in Expert systems, ed. Donald michie, Gordon &Breach, 1982.

RALAMBONDRAINY H.
"GENREG un générateur de règles combinant techniques d'apprentissage et techniques d'analyse de données", in (CEREMADE-87),1987.

RALAMBONDRAINY H.
"A symbolic numerical algorithm GENREG for generating rules from data" INRIA Report 1988 (in print)

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique