Pergamon

Computers chem. Engng Vol. 19, Suppl., pp. S615–S620, 1995

0098-1354(95)00128-X
0098-1354/95 $9.50 + 0.00

A FRAMEWORK FOR INVESTIGATING SCHEDULE ROBUSTNESS UNDER UNCERTAINTY

D.J. MIGNON*, S.J. HONKOMP**, G.V. REKLAITIS**@

* Process Engineering Department, Catholic University of Louvain-la-Neuve
1, Voie Minckelers, B-1348 Louvain-la-Neuve, BELGIUM

** School of Chemical Engineering, Purdue University
Chemical Engineering Building, West Lafayette, IN 47907-1283, USA

ABSTRACT

The batch processing manufacturing environment is subject to a high degree of uncertainty. One of the key issues in producing practical process schedules in such an unfavourable environment is the assessment of their robustness under uncertainty. This paper shows how Monte-Carlo based simulation models, implemented within the framework of the BATCHES simulator, can be used for this purpose, starting with the automated implementation of a given schedule into the simulation models, then running the various replicates of the simulation and finally analysing their results. In addition, the possibility of feeding back the simulation results to the scheduling model is also discussed, paving the way towards the development of a more effective reactive scheduling system.

KEYWORDS

Chemical Process, Batch Processes, Optimization, Scheduling, Simulation, Mixed Integer Linear Programming

INTRODUCTION

Uncertainty in batch processes can originate from demand or order changes, order priority changes, batch or equipment failures, processing time variability, resource changes and/or recipe variations. Batch process scheduling methodologies aim at producing feasible, robust and possibly optimal schedules. One of the key issues in producing such practical schedules is the assessment of their robustness in the presence of these various sources of uncertainty.

This paper shows how Monte-Carlo based simulation models, implemented within the framework of the BATCHES simulator, can be used for this assessment purpose. Monte-Carlo simulation methodology is indeed most appropriate for the representation of systems where stochastic variability plays a major role. It is shown how the different schedule parameters (order times, batch sizes, equipment preferences, batch splitting and merging, equipment and/or batch size dependent cycle times, task dependent changeover times, etc.) can be easily implemented in the simulation models. Consequently, since the simulation models can accommodate stochastic variability of numerous parameters, ranging from processing times to operators and utilities requirements, the analysis of simulation results gives good insights into the manner in

@ Author to whom correspondence should be directed

which the real process would perform under the schedule being tested, thus providing an assessment of the quality of the latter. This paper also shows how the procedure of implementing a given schedule (produced by an MILP scheduling package) into the simulation model, running the various replicates of the simulation and analyzing their results can be automated in order to make it as quick and efficient as possible.

Finally, this paper discusses the possibility of extending the use of this simulation framework to evaluate the performance of reactive scheduling algorithms. Indeed, by incorporating some degree of decision making during the execution of the simulation, this framework yields an adequate representation of an industrial production environment and its behaviour under real-time scheduling. We will briefly indicate how this idea can be implemented within the structure of a feedback system.

SIMULATION AND SCHEDULING: TWO WORLDS APART?

Conceptually, schedulers and simulators are very different. The scheduler generally represents a process as a mixed integer liner program, allowing the entire time horizon to be considered and an optimal solution to be produced for a deterministic scenario. A simulator, on the other hand, steps through time using local decision variables to determine when the next event is to take place, not concerned with the global optimality of a solution. Also, uncertainty is easily implemented into the simulation framework making it ideal for representing a real plant.

There have been efforts to integrate batch process simulation and scheduling in the past (Cott and Macchietto, 1989 b ; Clark et al., 1993), however these have, for the most part, been ad hoc schemes rather than direct integration. Indeed, these two technologies have generally tended to follow independent evolutionary paths. Each considers the same industrial reality from a different perspective and uses different simplifying assumptions in the representation of this reality. Consequently, the first step in the implementation of an integrated environment for simulation and scheduling is to make sure that both "partners" could adequately take into account various aspects of the process being considered. In our work, the BATCHES simulator (Clark and Joglekar, 1992) has proved to be quite flexible, and suitable for the development of simulation models representing most, if not all, features of a discrete optimization based scheduler, in our particular case, implemented in the RCSP++ system (Zentner et al., 1994). Table 1 shows the equivalence between the elements of a scheduling model and the way they have been implemented in the simulation model.

Table 1. Equivalence between scheduling and simulation entities.

RCSP++ SCHEDULING MODEL ELEMENTS	BATCHES SIMULATION MODEL ELEMENTS
Resource arrivals	Arrivals
Resource "free" purchases	Use of raw materials
Resource consumption and production by processing tasks	Tasks subdivided into detailed subtasks
Fixed task durations	Tasks and subtasks models
Equipment- and/or batch size-dependent processing times	User written functions, state or time event form
Batch splitting	Split subtask models
Batch merging	Input to filling subtasks with "grab-it flag" value = 2
Product demands	Products are stored in "infinite" capacity storage tanks and product demands are considered during the results analysis step

The "user written functions" mentioned in Table 1 consist of routines written in C or FORTRAN by the user, then compiled and linked with the main simulation executable. In these routines, the user specifies how some simulation parameters, such as the processing times or the utility requirements, must be computed on the basis of other parameters such as the batch size, the unit being used, the reaction kinetics, or the time of day at which the task is carried out. In general, any complex state dependent computation can readily be implemented.

For further information about these various elements of the scheduling and simulation models, the reader should refer to the references for the RCSP++ scheduling system (Zentner et al, 1994) and the BATCHES simulator (Clark and Joglekar, 1992).

Automatic Schedule Implementation

In the framework that is investigated here, a twin model of the process is thus developed, in the BATCHES simulator on the one hand, and in the scheduling environment (RCSP++) on the other hand. The model in RCSP++ is then used to generate a production master schedule which will hopefully be both optimal (this is guaranteed in the purely deterministic case) and robust (accommodating uncertainty). The simulation model in which stochastic variability of various parameters has been incorporated is used in turn to assess the quality of the schedule. In order to do so, the latter must be translated to the simulation environment, and this is performed by the GENSCHED program which reads information from the scheduler output file and converts it in the format of PPS's into the simulator input file. In addition, the schedule information is stored in memory at the beginning of the simulation so that it is available during the simulation run and can be used for reactive scheduling purposes.

Performance Assessment of a Simulation Run

A key question is how to measure the performance of the schedule imposed on the plant represented in the simulation model. The performance of a schedule during the scheduling step is generally assessed through the computation of a detailed profit based objective function. Therefore, in order to be fully consistent, we have reconstructed the value of the same objective function from the simulation results and are thereby able to determine its value for each realization of the uncertain parameters. A simple comparison of the two objective function values will provide an estimate of how well the schedule will perform in reality.

The computation of the same objective function at both the scheduling and the simulation levels is also very important in order to validate the simulation model. Indeed, the first step in validating the simulation model involves running the model with the mean values of the stochastic parameters. The objective function value for this run should match the objective function value obtained from the scheduler. Only after this validation step will the stochastic variations of some parameters be introduced.

Finally, in the case of reactive scheduling, the ability to compute the overall value of the objective function at the end of the simulation, after the master schedule may have been modified via rescheduling or other adjustments, will provide the designer with an ideal tool to assess the quality of both the master schedule and the reactive scheduling system.

Therefore, in order to compute the objective function value in the end of each simulation run, the following data are collected: raw materials consumption profiles, utilities and operator usage profiles, task and subtask execution profiles, and profiles of material levels in equipment units. Together with the simulation data, the cost information is then loaded into the program ANARESULT and the value of the objective function is computed. This value comprises the following terms:

- task execution costs and changeover costs (may be equipment-, batch size- and/or time-dependent);
- resource costs (raw materials, utilities, operators; may be time-dependent);
- storage costs (may be unit- and/or time-dependent);
- inventory value by the end of the time horizon;
- sales incomes and penalties for late deliveries, depending on the type of order.

As this list suggests, the information that can be obtained from this computation is fairly rich and gives a very clear picture of the simulated performance of the plant. Moreover, the comparison of the individual terms of the objective function values of different simulation runs can often be used to track and find the source of problems which caused a degradation of the process performance.

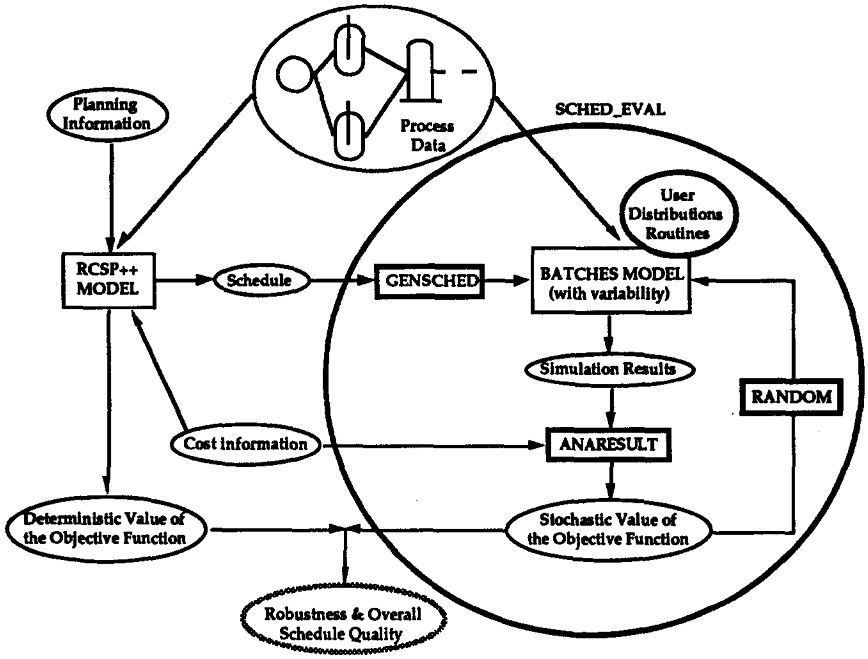

Figure 1. Components for the integrated system for schedule assessment

Putting Together the Pieces

It is well-known that, when dealing with process simulation involving stochastic variability of some parameters, statistically significant results can only be obtained if sufficient simulation replicates are run. With each replicate the values of the seeds of the pseudo-random number generators must be changed. This is done automatically in our environment by the small program RANDOM which is part of a loop carrying out the required number of simulation replicates. After each replicate, the simulation information is extracted and the value of the objective function is computed.

The complete structure of the integrated framework that has been described so far is presented in Fig. 1. In this figure, the different modules which have been developed during this study are outlined in the thick-bordered boxes. As illustrated by the larger box called SCHED_EVAL, the complete process of implementing a given schedule produced by RCSP++ within the simulation model, including running the various replicates of the latter and analyzing their results, has been automated in order to make it as quick and efficient as possible.

EVALUATION OF SCHEDULE ROBUSTNESS AND PERFORMANCE

Once Monte-Carlo simulations are completed using the framework we have developed, a measure of schedule "robustness" can be computed as a function of the variability of the objective function value. The lower this variability, the more robust the schedule is. Therefore, as a simple measure of the robustness, we have chosen the following formula where the standard deviation $\widehat{OF_{sto}}$ of the objective function value is weighed by the absolute value of the deterministic objective value so as to reduce the impact of different orders of magnitude of the objective function values. The robustness is expressed in percentages, with a value closer to 100% characterizing a more robust schedule:

$$\text{Robustness} = 100 - \left(\frac{\widehat{OF_{sto}}}{|OF_{det}|} * 100 \right) \quad [\%] \tag{1}$$

The schedule "performance" is expressed as a percentage of the deterministic objective function value. The closer to 100% this value is, the better the performance.

$$\text{Performance} = \frac{\overline{OF_{sto}}}{OF_{det}} * 100 \quad [\%] \tag{2}$$

For an example we looked at a variation of a well known case, BATCH4 (Sahinidis and Grossmann, 1991), which has eight tasks that are processed on six units producing six intermediate and four final products. Whereas in the original problem all process parameters are supposed to be fixed, uncertainty has been introduced here by considering processing time variations as the stochastic perturbation in the BATCHES simulation model. The stochastic times were modelled as normal distributions with an average of one and a standard deviation of one tenth. Table 2 contains the summarized results of five hundred replicate runs with three scheduling strategies. Strategy One follows the times specified by the deterministic RCSP++ schedule exactly. Strategy Two allows forward shifting of event times when the RCSP++ schedule does not contain a gap but the realization of the schedule in the simulation would create a gap (taking advantage of early finishes). Strategy Three, is a "robust" schedule that was generated by advancing the product due dates one time period (planning for lateness). One can see that, qualitatively, Strategy Three was best followed by Strategy Two, and then Strategy One.

Table 2. Results of Simulation Replicates.

	Key Values of the Objective Function				
Strategy	OF_{det}	$\overline{OF_{sto}}$	$\widehat{OF_{sto}}$	Performance	Robustness
One	60559	55657	2101.1	91.90%	96.53%
Two	60559	56215	2215.7	92.82%	96.34%
Three	60559	60226	30.5	99.45%	99.95%

OF_{det} = Deterministic objective function value, i.e. value observed with no stochastic variability.

$\overline{OF_{sto}}$ = Average value of the objective function for the replicate runs with stochastic variability.

$\widehat{OF_{sto}}$ = Standard deviation of the objective function for the replicate runs with stochastic variability.

CONCLUSIONS AND PERSPECTIVES

The integrated environment that has been described provides the machinery to join the seemingly separate worlds of scheduling and simulation. This framework allows us to examine the performance and robustness of a schedule that has been obtained via deterministic methods when it is executed in a stochastic plant.

This approach, executed in an on-line fashion, allows extensive sensitivity analysis to be done on a schedule. A schedule can be tested in a Monte-Carlo fashion to check that the implementation of such a schedule would not lead to unstable operation of a plant or prevent it from meeting its objectives. The stability of a schedule intuitively depends on its flexibility, hence on its robustness, and on the ability of the scheduler to adapt to changes. Ultimately, an approach to generating master schedules must be developed that can better handle the uncertainties. Such robust schedules will in turn make it easier for a reactive scheduling system to cope with changes and uncertainty.

Currently the framework is being extended for the evaluation of the performance of reactive scheduling systems. Our implementation of a feedback scheduling structure relies on a double simulation model, composed of two independent sets of units and product recipes, respectively. The first set represents the real industrial process, with stochastic parameters in order to represent uncertainty, while the second uses the mean values of the stochastic parameters. The results of the stochastic model are compared with the expected results of the deterministic schedule, represented by the deterministic part of the simulation model. The schedule can then be updated according to the deviation of the plant from the expected operation and any changes in the orders. With this framework we plan to explore the effectiveness of various reactive scheduling strategies.

REFERENCES

Clark S.M. and Joglekar G.S. (1992). Simulation Software for Batch Process Engineering, *NATO Advanced Study Institute on Batch Processing Systems Engineering*, Antalya, Turkey (Proceedings to be published by Springer Verlag).

Clark S., Harper P., Kavuri S. and Joglekar G. (1993). An Integrated System for the Operation of Beverage Manufacturing Processes, *FOCAPO Meeting*, Crested Butte, CO (USA).

Cott B.J. and Macchietto S. (1989 a). Minimizing the Effects of Batch Process Variability Using Online Schedule Modification, *Computers Chem. Engng*, 13, No 1/2, 105-113.

Cott B.J. and Macchietto S. (1989 b). An Integrated Approach to Computer-Aided Operation of Batch Chemical Plants, *Computers Chem. Engng*, 13, No 11/12, 1263-1271.

Cott B.J. and Macchietto S. (1989 c). Strategies for Operating Batch Plants Subject to Variability - A Performance Assessment, *Chem Eng Res Des*, 67, 593-599.

Morari M. and Zafiriou E. (1989). *Robust Process Control*, Prentice-Hall, Englewood Cliffs, New Jersey, USA.

Sahinidis N.V. and Grossmann I.E. (1991). Reformulation of Multiperiod MILP Models for Planning and Scheduling of Chemical Processes, *Computers Chem. Engng*, 15, No 4, 255-272.

Zentner M.G., Pekny J.F., Miller D.L. and Reklaitis G.V. (1994). RCSP++: A Scheduling System for the Chemical Process Industry, *PSE 94*, Kyongju, Korea.