209

SIMULATION BASED PERFORMANCE MEASUREMENT AND ANALYSIS: AN INDUSTRIAL APPLICATION

David DUVIVIER, Abdelhakim ARTIBA and Valérie DHAEVERS
Centre de Recherches et d'Études en Gestion Industrielle (CREGI)
Facultés Universitaires Catholiques de Mons
B-7000 Mons – Belgium
Phone: +32(0)65323456
e-mail: duvivier@fucam.ac.be

This paper deals with discrete-continuous problems of planning and scheduling nonpreemptable jobs. The need of reusability and modularity leads us to build a "generic" simulation and optimization framework, which is described in this contribution. The validation of our platform is presented in terms of an application to a real highly constrained discrete-continuous problem.

1. INTRODUCTION

Scheduling problems occur in a lot of application fields such as production, timetabling and computer systems. These problems have already been widely studied and modeled (Carlier and Chrétienne, 1988; Pinedo, 1995; Pinedo and Chao, 1999; Vernadat, 1999). Each problem has its own characteristics; it is much more complicated when tackling real industrial problems.

Starting from object-oriented techniques we are developing a "generic" Framework dedicated to simulation and optimization of real discrete-continuous planning and scheduling problems.

In this paper, a real problem, inspired by the Fontainunion plant (at Fontaine-Lévêque, Belgium), is considered. The production consists in high quality steel wires and strands. The process is composed of continuous and discrete operations. Planning is done by an ERP (Enterprise Requirement Planning), so we focus our description on the scheduling part of the problem. Our objective is to simultaneously minimize late orders (both in terms of number of late orders and average lateness) and makespan (i.e. the maximum completion time of jobs).

The sequel of this contribution is organized as follows. In section 2, our framework is presented. In section 3, the Fontainunion problem is described. In section 4, some results are presented. Finally, we conclude and discuss some perspectives of this work in section 5.

2. THE *PLANORDO* FRAMEWORK

The objective is to realize a "*generic*" Framework dedicated to simulation and optimization of discrete-continuous planning and scheduling problems. Actually, a *generic* model does not exist "*as is*" and we need to clearly define the scope of our models. In the described project, generalization of our results concerns generalized jobshop planning and scheduling problems as defined in section 3 below.

Our Framework uses several concepts issuing from software engineering (Zobrist and Leonard, 1997) and modeling techniques, such as *design patterns* (Gamma et al., 1995; Zobrist and Leonard, 1997; Brown et al., 1998; Vlissides, 1998) and *Unified Modeling Language* (Booch et al., 1999) in order to build reusable modularized tools. The basic aim of our framework, named *PlanOrdo*, is to provide a modeling and simulation environment to be used as a test-bed to compare the performances of different scheduling strategies. It is based on three main components:

- One (or two) database(s);
- A simulation-model;
- A graphical user interface.

PlanOrdo is able to work with one or two databases: it depends on the coupling level of PlanOrdo to the information system of the target enterprise. When using one database, all data are in the same database. When using two databases, one database contains PlanOrdo data; the other one contains enterprise information. Figure 1 presents an example of integrating PlanOrdo to an ERP using two databases.

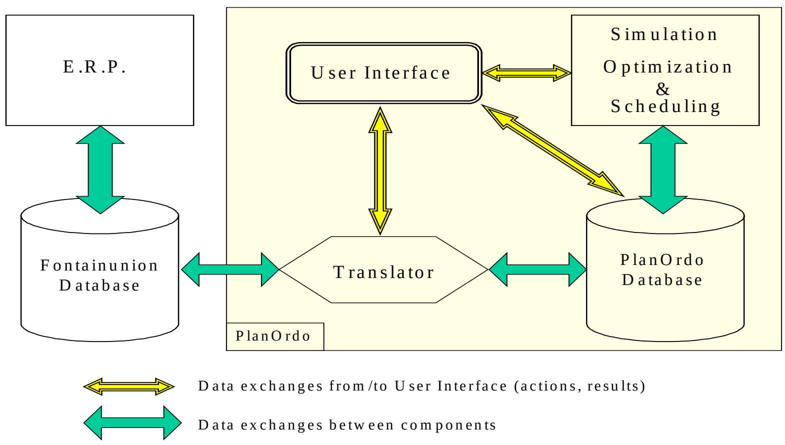

Figure 1 - PlanOrdo Overview

The simulation model is based on our RAO simulator (Artiba et al., 1998). Generally speaking, our RAO simulation model is composed of a generic set of patterns, and several specific patterns dedicated to the Fontainunion-problem.

The Graphical User Interface (GUI) is composed of several units described in next paragraphs. Up to now, the **scheduling unit** comprises a set of classical dispatching rules (such as Shortest Processing Time). These simple rules are not only easy/fast to implement, to control and to run, but also easy to explain to workers. An other advantage is that these rules apply to a wide range of scheduling problems. Major drawbacks of these rules is that they do not always give good results, and they are not stable when considering the variation of some parameters such as the load of the simulated process (Smith and Seidmann, 1983; Carlier and Chrétienne, 1988; Waikar et al., 1995; Hansmann and Hoeck, 1997). In current version of PlanOrdo, the **optimization unit** is reduced to a simple sorter (according to minimized/maximized criteria). We are working on a new metaheuristics-based version of this unit. The **performance measure unit** is able to generate a lot of statistics such as reports and Gantt diagrams. Several links to external spreadsheets and data plotting programs exists in order to extend this unit. The role of the **database translator** is to translate data from the information system of the enterprise to the PlanOrdo database and vice-versa. The **User Interface** is responsible for all graphical aspects of PlanOrdo. It's the supervisor of all units. The **RAO code generator** translates model parameters (issuing from the user interface) to the RAO language. **Interfaces to databases, Excel** & **Access-forms interfaces** and **RAO interface** are dedicated to data-exchange from/to PlanOrdo. Thanks to these units, PlanOrdo is independent to underlying database(s).

After defining the scope of our platform, we will describe the tackled problem in next section.

3. THE INDUSTRIAL PROBLEM

The validation of our approach is performed on the *Fontainunion* plant which is organized as a *generalized jobshop* composed of 17 machines divided in 6-to-7 stages (See Figure 2): in a *jobshop* organization every job has its own predefined route (*i.e.* it is processed on machines following its own order). In a *simple jobshop* (Carlier and Chrétienne, 1988; Jain and Meeran, 1999) each job has exactly one route and there is exactly one machine per stage. A *generalized jobshop* (JSP) is a jobshop where at least one stage is composed of more that one machine and/or at least one product may follow more than one route. At a low level of description, the shop comprises seven stages:

1. The first stage is the cleaning stage, it is composed of one machine;
2. The second stage is the wire drawing (stretching) stage, it is composed of five machines (called OTT1, OTT2, OTT3, BZ and HB);
3. The third stage is the stranding stage, it is composed of three machines (named STOL, SKIP, RED);
4. The fourth stage is the stabilization stage, its is composed of five machines (called TBR1, TBR2, TBR3, TBR4 and STOL);
5. The fifth stage is the packaging stage, it is composed of five machines (called TBR1, TBR2, TBR3, Redex and Frigerio);
6. The sixth stage is the sheathing stage, it is composed of one machine (Utifor);
7. Finally, the seventh stage is the packing stage, it is composed of two machines (called Emballage_Bottes and Emballage_Packs).

The production is divided in four main families:
- Monofil: simple wires;
- PT: small strands (external diameter less than 9.30mm);
- MT: medium strands (external diameter comprises between 9.30 and 11.30mm);
- GT: big strands (external diameter greater than or equal to 11.30mm).

Each product is composed of a sequence of operations that must be scheduled with respect to resource and precedence constraints. The resource constraints include input/output limitations and preemption (an operation cannot be stopped to start an other one). The precedence constraints state that no operation can begin before the end of all the preceding operations. The duration of an operation depends on the chosen machine. The production of the shop is not a cyclic production, demands variations are important from a month to another.

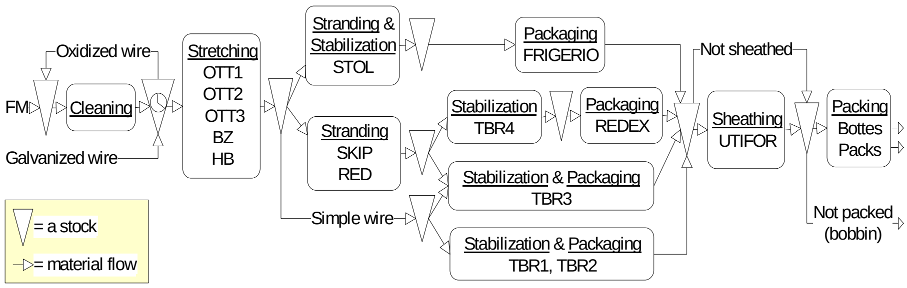

Figure 2 - Description of the Fontainunion-process

In this make-to-order process, several kinds of setups are considered (sequence-dependant, sequence-independent and weight-proportional setups). A lot of additional constraints make this problem harder, such as the use of bobbins (supporting products); or maximum/minimum input/output diameter, maximum input/output load; all kinds of bobbins are not compatible with all machines; all lines are not able to realize all products or packaging; some orders cannot be late; the model must cope with ruptures of wire and other malfunctioning aspects…

Holding costs and transport problems are not included because they have no significant impact on the process.

It should be noted that the continuous part of the problem is really easy to modelize using simple mathematical equations, so that it's possible to formulate the whole problem using our RAO discrete-event simulator (Artiba et al., 1998).

One of the original aspects of our work is the consideration of auxiliary resources. Considering Fontainunion-process, these resources are expressed in terms of bobbins. A scheduling strategy should avoid lacks of bobbins; moreover when considering the flexibility of lines, the strategy should keep – as far as possible – empty bobbins so as to consider uneven demands.

The schedule must satisfy several constraints such as:
1. Maximize throughput (in terms of tons/period);
2. Consider several levels of priority;

3. Fulfill due dates;
4. Minimize lead time, flow-time, makespan;
5. Insure load balancing of lines (capacity utilization);
6. Minimize setup times;
7. Maximize the flexibility of lines (as far as possible keep empty bobbins so as to consider uneven demands);
8. Minimize the use of temporary workers (and minimize week-ends overtime);
9. Minimize scrap.

When considering all these constraints it clearly appears that it might be really difficult to solve this problem with mathematical models, or even with simple metaheuristics. So we have chosen to use simulation to tackle the whole problem. Our approach consists in starting from a generic model and to complete this model step by step (continuous improvement approach). At the present time, our simulation-model is able to consider criteria 1 to 4 in the previous enumeration; we are still improving this model in order to consider all criteria. Before continuing the presentation, next paragraph introduces our notations (based on classical notations):

m is the number of machines	n is the number of jobs
d_j is the due date of job j	C_j is the completion time of job j
$E_j = \max(0, d_j - C_j)$ is the earliness of job j	$T_j = \max(0, C_j - d_j)$ is the tardiness of job j
N_E is the number of early jobs	N_T is the number of tardy jobs
$\overline{E} = \frac{1}{n}\sum_j E_j$ is the average earliness	$\overline{T} = \frac{1}{n}\sum_j T_j$ is the average tardiness
$C_{max} = \max_j C_j$ is the maximum completion time of jobs (called makespan)	

In this paper, the quality of a solution is measured in terms of several criteria:

(1) The first criterion is the throughput (in terms of tons/period);
(2) The second criterion is the number of tardy jobs N_T;
(3) The third criterion is the average tardiness $\overline{T}$;
(4) The fourth criterion is the makespan C_{max};
(5) The fifth criterion is the average earliness $\overline{E}$.

The objective is to minimize (2)-to-(4), and maximize (1)-and-(5). These criteria have been arbitrary chosen as an example, but PlanOrdo is not limited to these criteria. Next version of PlanOrdo will include a multi-objective performance unit. Its aim is to synthesize pertinent measures into scorecards.

Next section presents our results on a concrete example.

4. EXPERIMENTAL RESULTS

The objective of our model is to precisely schedule one month of production. On such a period, our model runs in less than three seconds on a Pentium IV 1.5GHz.

We investigate the impact of different scheduling strategies on studied criteria. Starting from simple dispatching rules, namely FIFO (First In First Out), SPT (Shortest Processing Time), LPT (Largest Processing Time), these rules have been extended to the price of product SPrix (Smallest Price) and LPrix (Largest Price) and combined to obtain 13 experimental scheduling strategies (D+DFT+FIFO, D+DFT+SPT, D+DFT+LPT, D+SPT, D+LPT, D+Sprix, D+Lprix, SPT+DFT+D, LPT+DFT+D, SPrix+DFT+D, LPrix+DFT+D, SPT+LPrix+DFT+D, D+FIFO).

Where D means starting date, and DFT is the due-date. The notation Criterion1+Criterion2+Criterion3 means that order lines are first sorted according to criterion1, criterion2 and finally criterion3.

Based on the 13 previously defined scheduling strategies, our results have been compared with real problem. In each case, PlanOrdo is able to obtain same results or even better results than those generated by the human planner (in terms of delays, makespan and lacks of bobbins).

According to confidentiality clauses related to this project, direct comparisons between simulation and reality are not presented. For the same reason, estimated orders might be used instead of real orders (but both are of same order of magnitude).

The following paragraphs present an example of utilization of PlanOrdo on the period ranging from 09/03/2001 to 10/11/2001. During this period 134 order-lines are simulated. At the end of the simulation, several performance measures are available (average load of machines, tonnages of products, Gantt charts…). Table 1 shows a comparison of the predefined strategies on the selected period. These strategies are sorted according to 5 criteria (from the most important to the less important one):

(1) Minimize the number of tardy jobs N_T (column N_T is sorted in ascending order);
(2) Minimize the average tardiness $\overline{T}$ (column $\underline{T}$ is sorted in ascending order);
(3) Minimize the makespan C_{max} (column *Cmax* is sorted in ascending order);
(4) Maximize the average earliness $\overline{E}$ (column $\underline{E}$ is sorted in descending order)
(5) Minimize the duration of the run (column SimDur is sorted in ascending order).

Criteria (1), (2), (3) and (4) have already been presented and discussed in section 3. The last criterion chooses the fastest strategy (in terms of computational speed) among strategies that have the same values of previous criteria.

Table 1 – Comparison of scheduling strategies

Nr	Strategy	SimDur (Sec)	Cmax (Hours)	N_T	$\overline{T}$ (Hours)	N_E	$\overline{E}$ (Hours)	Rank
1	D+DFT+SPT	2,31	757,71	0	0	134	277,4	5
3	D+SPT	2,31	757,71	0	0	134	277,4	5
5	D+Sprix	2,33	757,71	0	0	134	277,15	4
2	D+DFT+LPT	2,41	760,17	0	0	134	256,84	8
4	D+LPT	2,42	760,17	0	0	134	256,84	8
6	D+Lprix	2,4	760,17	0	0	134	256,83	10
12	D+FIFO	2,38	767,79	0	0	134	268,74	11
0	D+DFT+FIFO	2,4	767,79	0	0	134	268,74	11
7	SPT+DFT+D	2,3	749,77	2	17,65	132	287,42	2
11	SPT+Lprix+DFT+D	2,3	749,77	2	17,65	132	287,42	2
9	Sprix+DFT+D	2,3	750,8	3	23,09	131	288,62	1
10	Lprix+DFT+D	2,46	754,08	4	17,08	130	259,82	7
8	LPT+DFT+D	2,45	756,68	6	10,55	128	266,57	13

Considering the selected period and criteria, the best strategy is D+DFT+SPT. This result is coherent with results of other authors. It shows that the best strategy consists in respecting starting dates (computed by the human planner), then sorting order-lines according to due dates, and finally start shorter commands first (SPT dispatching rule).

It should be pointed out that these results, related to the selected period, must not be generalized to an other period due to the influence of several parameters on the performance of dispatching rules as mentioned in section 2.

Other performance measures are available in order to compare scheduling strategies. For example, PlanOrdo is able to provide several comparison diagrams based on the evolution of tonnage of finished goods, or on the use of bobbins.

PlanOrdo is able to plot the evolution of the use of bobbins, or to compute synthesized statistics about the lack of bobbins so as to obtain a classification of scheduling strategies according to the duration of the lack of bobbins (See column "Rank" in Table 1).

Considering the selected period, the three best strategies are (according to all previously defined criteria considering that the most important criteria is N_T and the less important one is the lack of bobbins) are 1, 3 and 5 (that is D+DFT+SPT, D+SPT, and D+Sprix).

These results show that the classification of strategies according to the two kinds of criteria (N_T, $\overline{T}$, C_{max} and $\overline{E}$ on one hand, and the lack of bobbins on the other hand) is not so evident. That's a reason why we want to complete the optimization module so as to include multi-objective scheduling coupled with a performance unit dedicated to the synthesis of pertinent measures into scorecards.

We have studied the behavior of our model on several periods; results remain coherent with those generated by the human planner (in terms of delays, makespan and lacks of bobbins).

5. CONCLUSIONS AND PERSPECTIVES

In this paper we have presented a simulation approach of a real-life industrial problem. We focus on the description of the rational of the process and its logic of functioning, pointing out the importance of the criteria embedded in the objective function. The obtained results are similar or even better that those generated by the human planner. Different scheduling strategies and performance measures have been described and compared. The results are consistent and show the robustness of the developed generic model. This means that the proposed approach is not restricted to the Fontainunion presented problem. We are still improving the Framework by integrating some multicriteria scheduling methods. We are working on a new version of the schedule-builder that includes a metaheuristic. In this new version, the simulation-model plays the role of an objective function.

5.1. Acknowledgments

This work is part of a project subsidized by the Walloon Region, projects Nr. 991/4274 and 991/4367.

4. REFERENCES

1. Artiba, A., Emelyanov, V. V., and Iassinovski, S. I. (1998). Introduction to Intelligent Simulation: The RAO Language. Kluwer Academic Publishers. ISBN 0-7923-8176-9.
2. Booch, G., Rumbaugh, J., and Jacobson, I. (1999). The Unified Modeling Language - User Guide. Addison Wesley. ISBN: 0-201-57168-4.
3. Brown, W. J., Malveau, R. C., McCormick, H. W., and Mowbray, T. J. (1998). Anti Patterns - Refactoring Software, Architectures and Projects in Crisis. Wiley Computer Publishing. ISBN: 0-471-19713-0.
4. Carlier, J. and Chrétienne, P. (1988). Problèmes d'ordonnancement, modélisation, complexité, algorithmes. Masson. ISBN: 2-225-81275-6.
5. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). Design Patterns CD - Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN: 0-201-30953-X (CDROM).
6. Hansmann, K.-W. and Hoeck, M. (1997). Production control of a flexible manufacturing system in a job shop environment. International Transactions in Operational Research, 4(5/6):341--351. ISSN: 0969-6016.
7. Jain, A. and Meeran, S. (1999). Deterministic job-shop scheduling: Past, present and future. European Journal of Operational Research, 113:390--434.
8. Pinedo, M. (1995). Scheduling - Theory, Algorithms, and Systems. Prentice Hall. ISBN: 0-137-06757-7.
9. Pinedo, M. and Chao, X. (1999). Operations Scheduling with applications in manufacturing and services. Irwin/McGraw-Hill. ISBN: 0-07-289779-1.
10. Smith, M. L. and Seidmann, A. (1983). Due date selection procedures for job-shop simulation. Computers and Industrial Engineering Journal, 7(3):199--207.
11. Vernadat, F. (1999). Techniques de Modélisation en Entreprise : Applications aux Processus Opérationnels. Economica. ISBN: 2-717-83853-8.
12. Vlissides, J. (1998). Pattern Hatching - Design Patterns Applied. Addison-Wesley. ISBN: 0-201-43293-5.
13. Waikar, A. M., Sarker, B. R., and Lal, A. M. (1995). A comparative study of some priority dispatching rules under different shop loads. Production Planning Control, 6(4):301--310. ISSN: 0953-7287.
14. Zobrist, G. W. and Leonard, J. V. (1997). Object-Oriented Simulation - Reusability, Adaptability, Maintainability. IEEE Press. ISBN: 0-780-31061-6.